

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ТЕПЛОЕНЕРГЕТИЧНИЙ ФАКУЛЬТЕТ
Кафедра автоматизації проектування енергетичних процесів і систем

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри

(підпис) О.В. Коваль
(ініціали, прізвище)
“ ” _____ 2019 р.

Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення
за спеціалізацією Інженерія програмного забезпечення розподілених систем
на тему: Обробка аудіосигналів нейромережевими методами

Виконав: студент 6 курсу, групи ТВ-81мп
(шифр групи)

Витвицький Дмитро Артурович
(прізвище, ім'я, по батькові) _____
(підпис)

Науковий керівник к.т.н., доцент Шаповалова С.І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____
(підпис)

Рецензент засуттник декана ТЕФ, доцент, к.т.н. Шавель Є.В.
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) _____
(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією – Інженерія програмного забезпечення розподілених систем

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) О.В. Коваль
” ” _____ 2019р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Витвицькому Дмитру Артуровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Обробка аудіосигналів нейромережевими методами

керівник роботи Шаповалова Світлана Ігорівна, к.т.н., доцент

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від 4 листопада 2019 р. № 3812-с

2. Строк подання студентом роботи _____ грудня 2019р.

3. Об'єкт дослідження загальний підхід до розв'язування задач обробки аудіосигналів на основі методів штучного інтелекту.

4. Предмет дослідження розпізнавання звучання музичних інструментів на основі нейронної мережі прямого поширення та згорткової нейронної мережі.

5. Перелік питань, які потрібно розробити провести аналіз попередніх досліджень та сучасних методів вирішення задачі класифікації музичних інструментів; запропонувати шаблонне рішення для розв'язання задачі обробки аудіосигналів; підібрати дані для навчальної вибірки для нейронної мережі; спроектувати архітектуру нейронної мережі; розробити класифікатор музичних інструментів; виконати візуалізацію вхідних даних, процесу навчання нейронної мережі та результатів.

6. Орієнтовний перелік ілюстративного матеріалу: вхідні та вихідні дані для вирішення задачі розпізнавання музичних інструментів, типи музичних інструментів, методи розпізнавання музичних інструментів, загальне представлення до вирішення задачі, архітектура програмного додатку, архітектура багат шарової нейронної мережі прямого поширення використаної в даному досліді, архітектура згорткової нейронної мережі використаної в даному досліді, характеристики музичного

інструменту на прикладі віолончелі, графік представлення тестової вибірки у двовимірному просторі, отриманого методом виділення головних компонент, графіки навчання, матриця колізій отриманих результатів, порівняння якості розпізнавання для нейронної мережі прямого поширення та згорткової нейронної мережі та приклад виконання програми.

7. Орієнтований перелік публікацій Архітектура нейронної мережі для обробки аудіосигналів (“Сучасні проблеми наукового забезпечення енергетики” XVII міжнародної науково-практичної конференції аспірантів, магістрів, студентів)

8. Дата видачі завдання ”18” вересня 2019 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	28.09.18 р.	
2	Опрацювання літературних джерел	01.10.18 р. – 03.02.19 р.	
3	Підготовка матеріалів дисертації	04.02 – 31.05.19 р.	
4	Підготовка доповідей на конференції	11.03 – 29.03.19 р.	
6	Розробка програмного продукту	03.06 – 25.10.19 р.	
5	Переддипломна практика	02.09 – 25.10.19 р.	
7	Захист програмного продукту	26.10.19 р.	
8	Розробка стартап-проекту	11.11 – 19.11.19 р.	
9	Передзахист	20.11.19 р.	
10	Оформлення дисертації	21.11- 29.11.19 р.	
11	Захист	16.12.19 р.	

Студент

(підпис)

Витвицький Д.А.

(прізвище та ініціали,)

Керівник роботи

(підпис)

Шаповалова С.І.

(прізвище та ініціали,)

РЕФЕРАТ

Структура і обсяг дипломної роботи

Магістерська дисертація складається зі вступу, шести розділів і висновку. Робота містить в собі 33 джерела за переліком посилань, 35 ілюстрацій, 25 таблиць. Крім того, в кінці розміщено 1 додаток. Основна частина роботи викладена на 92 сторінках.

Актуальність теми

Пошук та аналіз музичної інформації має широке практичне застосування в задачах автоматизації процесів обробки звукових сигналів. До таких задач належать автоматичне наповнення баз даних, анотація звукових сигналів, розпізнавання мелодії на основі уривку, розпізнавання музичних інструментів, оцінка якості синтезованого звуку та інші. Комплексні проблеми в обробці звукових сигналів визначають актуальність вирішення окремих підзадач, зокрема класифікації музичного інструменту за його звучанням. Використання методів машинного навчання в задачах класифікації музичних інструментів має потенціал з точки зору підвищення ефективності систем обробки музичної інформації.

Дана робота присвячена розробці шаблонного рішення для задач обробки аудіосигналів, на прикладі розв'язання задачі класифікації.

Метою дослідження є створення шаблонного рішення для розв'язання задач обробки аудіосигналів, на прикладі задачі класифікації звучання музичних інструментів.

Для досягнення поставленої мети потрібно виконати наступні **задачі дослідження**:

- розглянути та проаналізувати попередні дослідження в сфері обробки аудіосигналів;

- на основі проведеного аналізу виділити загальні кроки для вирішення задачі класифікації та сформулювати шаблонне рішення для розв'язання задач класифікації аудіосигналів;

— дослідити музичні характеристики, їхнє використання в попередніх роботах та вплив на якість класифікації;

— розглянути програмні засоби для вирішення задач екстракції музичних характеристик, побудови нейронної мережі та візуалізації отриманих даних;

— створити архітектури для глибинної нейронної мережі прямого поширення та згорткової нейронної мережі, реалізувати розроблену архітектуру за допомогою програмних засобів;

— провести тренування та тестування моделей штучного інтелекту та провести порівняння для двох методів;

— на основі рішень попередніх задач реалізувати програмний продукт у вигляді веб-застосунку.

Об'єкт дослідження: загальний підхід до розв'язування задач обробки аудіосигналів на основі методів штучного інтелекту.

Предмет дослідження: розпізнавання звучання музичних інструментів на основі нейронної мережі прямого поширення та згорткової нейронної мережі.

Методи дослідження: методи штучного інтелекту, а саме нейронні мережі, для розпізнавання аудіосигналів, методи екстракції музичних характеристик.

Наукова новизна: набуло подальшого розвитку використання нейронних мереж для задач розпізнавання аудіосигналів, а особливо згорткових нейронних мереж.

Практичне значення: запропонований підхід до вирішення розв'язання задач розпізнавання аудіосигналів дозволяє спеціалістам легше розпочати роботу з новими методами до вирішення подібних задач, тобто зменшується вхідний поріг для нових користувачів. Розробникам нейронних мереж, які раніше працювали з обробкою зображень, продемонстровано загальні етапи для вирішення задачі виділення звукових характеристик.

Апробація: результати досліджень стосовно цілої системи взагалом представлені на XVII міжнародній науково-практичній конференції аспірантів, магістрів, студентів на тему «Сучасні проблеми наукового забезпечення енергетики» 2019 року.

Ключові слова: НЕЙРОННА МЕРЕЖА, КЛАСИФІКАТОР, МУЗИЧНИЙ ІНСТРУМЕНТ, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, НЕЙРОННА МЕРЕЖА ПРЯМОГО ПОШИРЕННЯ, ШТУЧНИЙ ІНТЕЛЕКТ.

ABSTRACT

Structure and volume of the thesis

The master's thesis consists of an introduction, six sections and a conclusion. The work contains 33 sources in the list of references, 35 illustrations, 25 tables. In addition, there is 1 app at the end. The bulk of the work is set out on 92 pages.

Actuality of theme

Searching for and analyzing music information has wide practical application in the tasks of automation of audio signal processing. Such tasks include automatic filling of databases, annotation of sound signals, recognition of a tune based on a passage, recognition of musical instruments, evaluation of the quality of synthesized sound and others. Complex problems in the processing of sound signals determine the urgency of solving individual sub-tasks, in particular the classification of a musical instrument by its sound. The use of machine learning methods in the tasks of classifying musical instruments has the potential in terms of improving the efficiency of musical information processing systems.

This paper is devoted to the development of a template solution for audio processing problems, for example the solution of the classification problem.

The purpose of the study is to create a template solution for solving audio processing tasks, such as the task of classifying the sound of musical instruments.

To achieve this goal, you need to complete the following **research objectives**:

- review and analyze previous research in the field of audio signal processing;
- to identify, on the basis of the analysis, the general steps for solving the classification problem and to form a template solution for solving the problems of classification of audio signals;
- to study musical characteristics, their use in previous works and influence on the quality of classification;

- consider software tools for solving the problems of extracting musical characteristics, building a neural network and visualizing the data obtained;
- to create architectures for deep neural network of direct propagation and convolutional neural network, to implement the developed architecture by means of software;
- train and test artificial intelligence models and compare for two methods;
- to implement the software product in the form of a web application based on the solutions of previous problems.

Object of study: A general approach to decoupling audio processing tasks based on artificial intelligence methods.

Subject of study: recognition of the sound of musical instruments on the basis of neural network direct propagation and convolutional neural network.

Research Methods: Artificial intelligence methods, namely neural networks, for audio signal recognition, methods for extracting musical characteristics.

Scientific novelty: The use of neural networks for the purpose of recognizing audio signals, and in particular convolutional neural networks, has been further developed.

Practical value: The proposed approach to solving audio recognition problems makes it easier for professionals to get started with new methods to solve similar problems, meaning the entry threshold for new users is reduced. Neural network developers who have previously worked with image processing have been shown the general steps to solve the problem of audio characteristics.

Testing: the results of research on the whole system are generally presented at the 18th international scientific conference of graduate students, masters, students on the topic "Modern problems of scientific support of energy" in 2019.

Keywords: NEURAL NETWORK, CLASSIFIER, MUSICAL INSTRUMENT, OBJECT RECOGNITION, CONVOLUTIONAL NEURAL NETWORK, ARTIFICIAL INTELLIGENCE

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	10
Вступ	11
1 Постановка задачі.....	13
2 Підходи до вирішення задачі класифікації музичних інструментів	16
2.1 Огляд попередніх робіт	16
2.2 Класичні методи розв’язання задачі	19
2.3 Сучасні підходи до розв’язання задачі	19
2.4 Засоби візуалізації навчальної вибірки.....	25
Висновки до розділу 2	27
3 Характеристики звучання.....	28
3.1 Часові характеристики.....	28
3.2 Спектральні характеристики.....	31
3.3 Обрані характеристики для дослідження	35
Висновки до розділу 3	35
4 Засоби розробки	36
4.1 Бібліотеки музичних записів	36
4.2 Засоби екстракції музичних характеристик	40
4.3 Засоби розробки нейронних мереж.....	41
Висновки до розділу 4	54
5 Програмна реалізація.....	56
5.1 Архітектура програмного додатку	56
5.2 Формування навчальної та тренувальної вибірки	57
5.3 Вирішення задачі за допомогою MLP	59
5.4 Вирішення задачі за допомогою CNN	62
5.5 Отримані результати.....	63
5.6 Інтерфейс користувача	66
Висновки до розділу 5	69
6 Розробка стартап проЕкту	70

6.1 Опис ідеї проекту	70
6.2 Технологічний аудит ідеї проекту.....	72
6.3 Аналіз ринкових можливостей запуску стартап-проекту	74
6.4 Розробка ринкової стратегії проекту.....	81
6.5 Розроблення маркетингової програми	84
Висновки до розділу 6	86
Висновки	88
Список використаних джерел	89
Додаток А.....	92

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HM (англ. NN – neural network)	нейронна мережа
FFT – Fast Furier transform	швидке перетворення Фур'є
MFCC – mel-frequency cepstral features	мел-частотні кепстральні хар-ки
PCA – principal component analysis	метод головних компонент
DRS – database retrieval systems	системи наповнення баз даних
IDE – integrated development environment	інтегроване середовище розробки
TF – TensorFlow	
API – application programming interface	прикладний програмний інтерфейс
GPU – graphical processor unit	графічний процесор
DL – deep learning	глибинне навчання
SOM – self-organizing map	
RBF – radial basis function	
QDA – quadratic discriminant analysis	
LSTM – long short-term memory	
MLP – multi-layer perceptron	
RMS – root mean square	
ReLU – rectified linear unit	
t-SNE – t-distributed stochastic neighbour embedding	t-розподілене вкладення стохастичної близькості
JSON – JavaScript Object Notation	

ВСТУП

Останнім часом спостерігається стрімкий розвиток досліджень в напрямку створення систем штучного інтелекту. Це зумовлено, в першу чергу, значними досягненнями в застосуванні нейронних мереж для обробки графічної інформації в задачах ідентифікації, класифікації, детекції. Ефективність алгоритмів машинного навчання історично частіше порівнюється на тестових задачах, пов'язаних з обробкою зображень. Створюються спеціалізовані репозиторії для навчальних вибірок зображень за різними класами, такі як MNIST чи CIFAR. Проводяться конкурси серед дослідників за різними уніфікованими тестовими задачами. В той же час, необхідність опрацювання надвеликих об'ємів аудіо-даних та вирішення проблеми створення так званого комп'ютерного слуху визначає актуальність досліджень щодо використання нейронних мереж в процесі аналізу звукового сигналу.

Пошук та аналіз музичної інформації має широке практичне застосування в задачах автоматизації процесів обробки звукових сигналів. До таких задач належать автоматичне наповнення баз даних, анотація звукових сигналів, розпізнавання мелодії на основі уривку, розпізнавання музичних інструментів, оцінка якості синтезованого звуку та інші. Комплексні проблеми в обробці звукових сигналів визначають актуальність вирішення окремих підзадач, зокрема класифікації музичного інструменту за його звучанням. В той же час, розвиток систем штучного інтелекту сприяв новим розробкам щодо застосування штучних нейронних мереж в галузі обробки аудіосигналів. Існує тенденція щодо розвитку спеціалізованих репозиторіїв музичних зразків, як потенційних баз даних для створення навчальних вибірок. Пропонуються нові підходи до обробки аудіо-інформації на базі нейронних мереж різного типу та архітектури. Використання методів машинного навчання в задачах класифікації музичних інструментів має потенціал з точки зору підвищення ефективності систем обробки музичної інформації.

Дана робота присвячена розробці шаблонного рішення для задач обробки аудіосигналів, на прикладі розв'язання задачі класифікації.

В роботі проведено огляд попередніх досліджень щодо підходів до класифікації музичних інструментів методами машинного навчання (розділ 2). Проаналізовано характеристики звукового сигналу, що можуть представляти набори значимих ознак для класифікації музичних інструментів (розділ 3). На основі порівняльного аналізу програмних фреймворків для побудови моделі штучної нейронної мережі обрано бібліотеку TensorFlow. Розроблено архітектуру програмного застосунку класифікації музичних інструментів з обґрунтуванням засобів реалізації. (розділ 4). На основі розробленої архітектури створено класифікатори з використання штучної нейронної мережі прямого поширення та згорткової нейронної мережі. Проведено обчислювальні експерименти, на основі запропонованих рішень (розділ 5).

Результати дослідження дозволяють зробити висновок щодо перспективності використання отриманої архітектури в задачах розпізнавання аудіосигналів.

1 ПОСТАНОВКА ЗАДАЧІ

Впродовж останніх років відбувається стрімкий розвиток технологій машинного навчання, які знайшли своє застосування в сферах обробки зображень та відео. Основні напрямки створення комп'ютерів з використанням штучного інтелекту пов'язані з вивченням візуальних даних. Сфера розпізнавання зображень та іншої візуальної інформації дуже важлива. Можливість обробляти велику кількість даних у певному контексті принесе велику цінність. Одним із прикладів є автоматизований процес транскрипції тексту, який передбачає застосування аналізу старовинних документів, з огляду на складність та нерегулярність записів внаслідок ручних аспектів написання. Було б чудово отримати доступ до онлайн ресурсів старих документів у веб-бібліотеках. Загальновідомо, що робота з арабського оптичного розпізнавання тексту переживає повільний процес розвитку порівняно з іншими мовами. З іншого боку, останні терористичні події змусили уряди збільшити інвестиції у сферу розпізнавання обличчя для підвищення безпеки. Дуже корисно застосовувати новітні технології в області розпізнавання обличчя для визнання людей за допомогою камери спостереження, встановленої в торгових центрах, аеропортах, університетах та компаніях. Глибинне навчання – це нове застосування машинного навчання для вивчення представлення даних. Наразі ця сфера переживає розквіт, але зазвичай в сферах розпізнавання зображень. Архітектури, засновані на DL, перемогли у багатьох викликах та змаганнях на веб-сайті Kaggle, які вирішують проблеми реального світу.

Однак, створення штучного інтелекту неможливе без аналізу звукових сигналів. Існує багато класичних методів розв'язання задач пов'язаних з дослідженням аудіо, однак зі зростанням кількості інформації ці методи погіршують свої результати. В той же час сучасні методи, що базуються на використанні нейронних мереж, на великий обсягах даних покращують свої показники, оскільки етапами побудови моделі є підбір тренувальної вибірки та процес навчання.

Основною перешкодою переходу до використання машинного навчання для обробки аудіосигналів слугує складність цього процесу, для спеціалістів необхідно

вивчати дві окремі галузі – обробки звуків та створення моделей штучного інтелекту. Отже, актуальною є задача переходу від класичних до сучасних методів розв’язання звукових задач. Для зменшення впливу даної проблеми пропонується створити загальне рішення для обробки аудіосигналів нейромережевими методами, яке допоможе спеціалістам швидше адаптуватись до нових технологій.

Автоматичне розпізнавання джерела звуку відіграє важливу роль у розробці автоматичної індексації і додатків для пошуку баз даних. Ці програми мають потенціал у порятунку людей, здійснюючи пошук через величезну кількість цифрових аудіо-матеріалів, доступних сьогодні. Наприклад, було б дуже корисно, мати змогу знайти звукові зразки, які «звучать схоже», як подано аудіо приклад. Аналіз музичного вмісту в цілому має багато практичних застосувань, включаючи, наприклад, структуроване кодування, автоматичне коментування музичного сигналу та інструментів музикантів. Автоматичне розпізнавання музичних інструментів є найважливішим завданням у вирішенні цих непростих проблеми, а також можуть надавати корисну інформацію в інших областях розпізнавання джерел звуку. Однак аналізу музичного сигналу так і не вдалося досягти великого комерційного інтересу, як, наприклад, розпізнавання мовців та мови. Це тому, що теми навколо мовленнєвої обробки є більш комерційно застосовними, хоча обидві області вважаються дуже складними. Через побудову комп’ютерних систем, які «Слухають», ми також можемо отримати нові уявлення про людське сприйняття. Ця теза описує побудова та оцінка системи розпізнавання музичних інструментів, яка здатна розпізнають поодинокі тони, що грають на оркестрових інструментах. [1]

Вирішення задачі поліфонічного звуку представлено в роботі. [2] Однак, результати за різних конфігурацій склали менше 62%. Для покращення отриманих результатів пропонується спочатку виділити джерело звуку, а потім відфільтровані дані моно-запису аналізувати за допомогою класифікатору звучання музичних інструментів. Оскільки процес виділення джерела звучання з аудіо-запису є окремою задачею, для спрощення пропонується вирішити задачу створення класифікатору звучання музичних інструментів нейромережевими методами.

До обробки аудіосигналів відносять задачі класифікації, музичного маркування, сегментації, виділення джерела звучання, відстеження ритму, підбір рекомендацій, виокремлення сигналів та транскрипції звукової доріжки. Для розв'язання подібних завдань нейромережевими методами необхідно вирішити декілька підзадач, а саме:

- вибір бібліотеки зразків для тренувальної вибірки;
- підбір найважливіших звукових характеристик;
- виокремлення обраних характеристик зі звукових сигналів;
- створення архітектури на тренування моделі нейронної мережі;
- перевірка отриманої моделі на тестовій вибірці.

Для прикладу, в даній роботі пропонується обрати для розв'язку задачу розпізнавання звучання музичних інструментів.

2 ПІДХОДИ ДО ВИРІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ МУЗИЧНИХ ІНСТРУМЕНТІВ

Зацікавленість в класифікації музичних інструментів істотно з'явилась ще в середині 90-тих років минулого століття. Традиційно розпізнавання музичних інструментів в основному базується на частотному аналізі та аналізі форм, щоб отримати набір різноманітних ознак.

У даному розділі розглянуто підходи та методи до вирішення задачі класифікації музичних інструментів.

2.1 Огляд попередніх робіт

Дослідження щодо розпізнавання музичних інструментів методами машинного навчання проводилися з середини 1990 років. Було створено різноманітні класифікатори, в тому числі на базі нейронних мереж, методу опорних векторів [3], методу К-найближчих сусідів [4], карт самоорганізації [5] та RBF [6]. В даних роботах використовувалися різні набори музичних інструментів, їх характеристик та бібліотек музичних записів для навчальної вибірки. Деякі з робіт представляють аналіз великої кількості музичних інструментів, проте класифікація виконується лише за сімействами. В інших роботах представлена класифікація в межах одного сімейства музичних інструментів.

Коректна класифікація музичних інструментів залежить від наступних виокремлених процесів:

- вибору музичних інструментів для розпізнавання;
- вибору характеристик;
- математичного представлення характеристик в наборі даних;
- зменшення розмірності набору даних;
- використання методів класифікації.

На початку експерименти проводились для малої кількості інструментів та характеристик. З часом удосконалювались методи та підходи до вирішення задачі і

кількість інструментів у експериментах зростала. Список експериментів приведено нижче (таблиця 2.1).

Таблиця 2.1. Підходи до вирішення задачі розпізнавання музичних інструментів

Експеримент	Інструменти	Приклади	Класифікатор	Точність (%)
Kaminskyj and Materka (1995)	4	240	PCA & MLP PCA & kNN	97.7 98.1
Cemgil and Gurgun (1997)	10	480	SOM-RBF	94.17
Fragoulis et al. (1999)	5	60	ARTMAP	83
Eronen (2001)	29	5286	k-NN	77
Eronen (2003)	27	5895	HMM	68
Martin and Kim (1998)	14	1023	k-NN Fisher	67.5 71.6
Martin (1999)	27	1500+	Hierarchy	75.9
Fujinaga (1998)	23	1338	GA & k-NN	50
Fraser and Fujinaga (1999)	23	1338	GA & k-NN	64
Marques and Moreno (1999)	8	1024	GMM SVM	63 70
Agostini et al. (2003)	27	1007	QDA SVM k-NN CDA	92.2 88.7 90.4 90
Kitahara et al. (2003)	19	6247	Bayes	79.73
Windmark, Toghiani-Rizi (2017)	30	1244	MLP	80
D. Kim (2018)	10	6705	CNN	77
P. Singh et. al. (2019)	10	9579	CNN SVM	50 78
Y. Hung (2018)	11	1000000	CNN	93
M. Yun (2017)	14	2800	RNN (LSTM) CNN (LeNet)	81.9 83

З переліку експериментів, описаних вище, видно, що багато різних методів окремо та їхнє сполучення були використані в спробі створити надійний класифікатор музичних інструментів. Тим не менше, жоден метод або техніка не

визнані найкращим методом для цього завдання. Це покладається на те, що результати можуть бути упередженими в деякий спосіб – наприклад, у виборі інструмента або виборі навчальних прикладів. Зразки записів гри музичних інструментів, що використовуються вище включають в себе MUMS, IRMAS, Університету Айови (Fritts, 1997), SHARC (Sandell, 1994). Кількість класифікованих інструментів коливається від 2 до 40 (з широким діапазоном окремих примірників кожного) та кількість характеристик коливається від 1 до сотні.

З розвитком технологій, вдосконалювались і методи машинного навчання, і в останні роки ця галузь стала однією з передових у сфері розпізнавання об'єктів. В останні роки запит на використання методів машинного навчання постійно зростає (рисунок 2.1) [24] і регулярно з'являються нові удосконалені інструменти для роботи з цими методами.

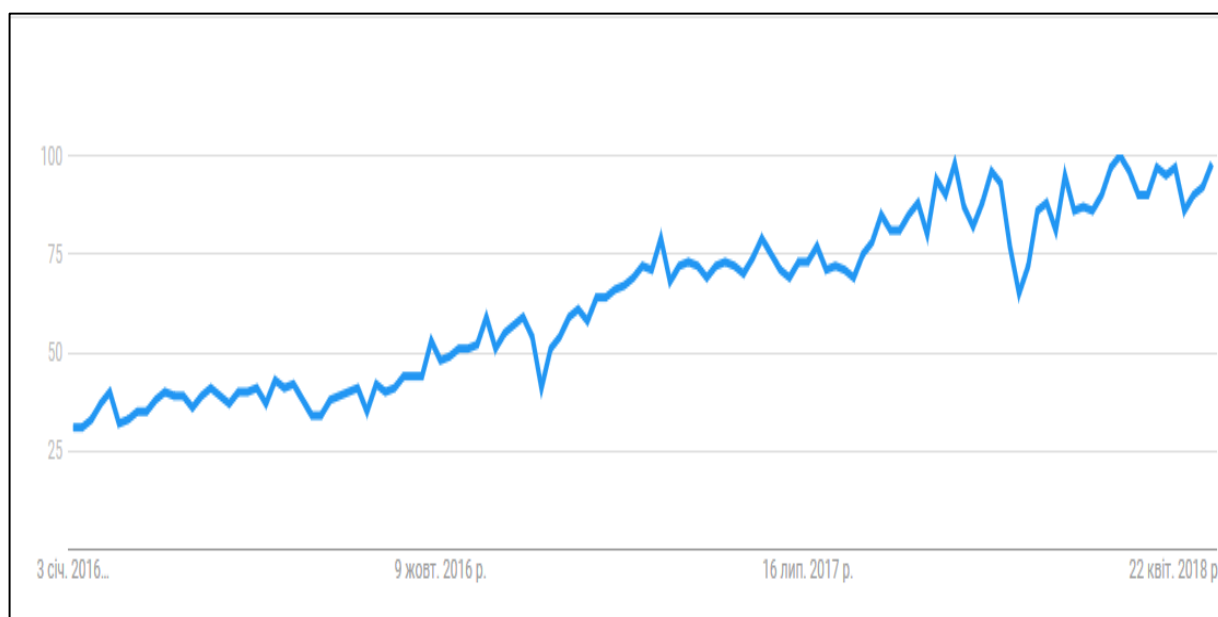


Рисунок 2.1 — Пошукові запити за темою машинного навчання з 2016 року

Методи вирішення задачі класифікації музичних інструментів поділяють на два типи – класичні та сучасні. В середині 90-х використовувались класичні методи, проте останнім часом з розвитком техніки почали застосовуватись сучасні методи класифікації з різними комбінаціями, у тому числі й поєднанням сучасних та класичних методів.

2.2 Класичні методи розв'язання задачі

До класичних підходів до вирішення задачі розпізнавання музичних інструментів відносять методи опорних векторів, виділення головних компонент та кластеризацію методом k-головних.

Метод опорних векторів – це набір алгоритмів навчання з вчителем, що використовується для задач класифікації та регресивного аналізу. Основна ідея полягає у створенні нового простору більш високої розмірності і пошуку роздільної гіпер-простору з максимальним зазором у цьому просторі. Ці площини будуються розділяючи елементи вибірки на декілька класів. [1]

Кластеризація методом k-головних полягає у впорядкуванні множини в порівняно однорідні групи. Мета методу полягає у розділенні n спостережень на k кластерів, так щоб кожне значення було частиною кластера з найближчим до нього середнім значенням. Метод базується на зменшенні суми квадратів відстаней між кожним спостереженням та центром його кластера.

2.3 Сучасні підходи до розв'язання задачі

Одним з методів вирішення задачі розпізнавання музичних інструментів який відносять до сучасних методів є нейронні мережі. Нейронні мережі – це системи які працюють на зразок людського мозку. Вони складаються з нейронів, які є сукупністю з'єднаних вузлів. Кожне з таких з'єднань може переносити інформацію. Нейрон, отримуючи сигнал, обробляє його, після чого подає сигнал нейронам приєднаним до нього. Важливою складовою нейронної мережі є функція поширення та правило навчання. Функція поширення обчислює вхід до нейрону з виходів нейронів попередників, а правило навчання – це алгоритм, що підлаштовує параметри нейронної мережі, який полягає у зміні ваг та порогів змінних мережі.

Існує декілька типів нейронних мереж, а саме зворотного поширення, рекурентні, глибинні, згорткові, тензорні глибинні складальні та багато інших.

Багатошаровий персептрон (MLP) – це клас підсилювальної штучної нейронної мережі.

MLP складається з щонайменше трьох шарів: вхідного, прихованого та вихідного (рисунок 2.2). [8] За винятком вхідних вузлів, кожен вузол є нейроном, який використовує функцію нелінійної активації. MLP використовує керовану методику навчання, яку називають метод зворотного поширення помилки (backpropagation).

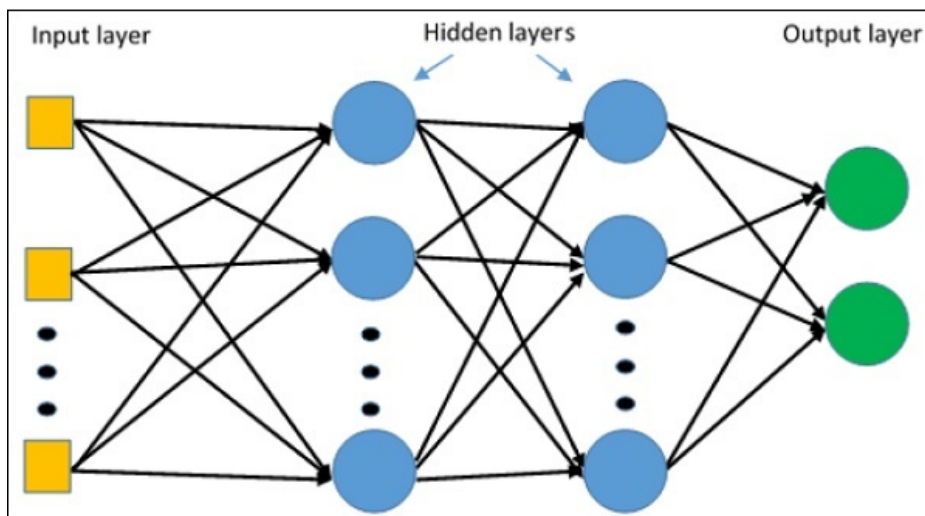


Рисунок 2.2 — Архітектура багатошарового персептрону

Головною перевагою персептрону є те, що він простий для розуміння і вивчення, не потрібно довго розбиратись з його архітектурою, можна за короткий проміжок часу побудувати просту нейронну мережу, яка буде показувати відносно хороші результати. Його слід використовувати для регресії та картографування. Він може використовуватися для відображення N-мірного вхідного сигналу на M-вимірний вихідний сигнал, таке відображення також може бути нелінійним.

Основне обмеження алгоритму MLP полягає в тому, що через те, як він навчається, він не може гарантувати, що мінімуми, на яких він зупиняється під час тренування, є глобальними мінімумами. Отже, алгоритм MLP може застрягнути в локальних мінімумах. Один із варіантів пом'якшення цього – це кілька разів тренувати алгоритм MLP, використовуючи кожен раз інше випадкове вихідне положення, а потім вибрати модель, що призводить до найменшої помилки RMS.

Встановлення більшої кількості випадкових ітерацій тренувань, може призвести до кращої моделі класифікації або регресії, однак це збільшить загальний час навчання. Ще одне обмеження алгоритму MLP полягає в тому, що кількість прихованих нейронів повинно встановлюватись користувачем, якщо встановити це значення занадто низько, це може призвести до недосконалості моделі MLP, тоді як встановлення цього значення буде занадто високим, це може призвести до перенавчання моделі MLP.

Однією з найактуальніших робіт у сфері розпізнавання звучання аудіосигналів за допомогою MLP є стаття. [9] В ній автор пропонує вирішити задачу розпізнавання звучання музичних інструментів. В архітектурі нейронної мережі використовуються 50 мел-кепстральних музичних характеристик на вході, 30 вузлів в прихованому шарі та 8 вихідних вузлів, кожен з яких представляє конкретний інструмент. При цьому в роботі аналізується також вплив атаки інструменту на якість розпізнавання. Експеримент досягає результату у 80% для класифікації окремого інструменту.

Згорткова нейронна мережа (CNN) складається з одного або декількох згорткових шарів, за якими слідує один або більше повністю пов'язаних шарів, як у стандартній багатошаровій нейронній мережі (рисунок 2.3). [10]

Архітектура CNN створена для того, щоб скористатися 2D структурою вхідного зображення (або іншого 2D входу, наприклад аудіосигналу). Ще одна перевага CNN – це те, що їх легше тренувати і мають набагато менше параметрів, ніж повністю пов'язані мережі з однаковою кількістю прихованих шарів.

Вхід до згорткового шару – це зображення $m \times m \times r$, де m – висота і ширина зображення, а r – кількість каналів, наприклад. RGB-зображення має $r = 3$.

Згорнутий шар матиме k -фільтри розміром $n \times n \times q$, де n менше розмірності зображення, а q може бути таким самим, як кількість каналів r або меншим, і може змінюватись для кожного фільтра. Розмір фільтрів породжує локально пов'язану структуру, кожен з яких поєднано із зображенням, щоб створити k -карти розмірів $m - n + 1$. Потім для кожної карти застосовується підбір проб (subsampling), як правило, із середнім або максимальним об'єднанням (average or max pooling) для суміжних

областей $p \times p$, де p коливається між 2 для невеликих зображень і зазвичай не більше 5 для більших входів.

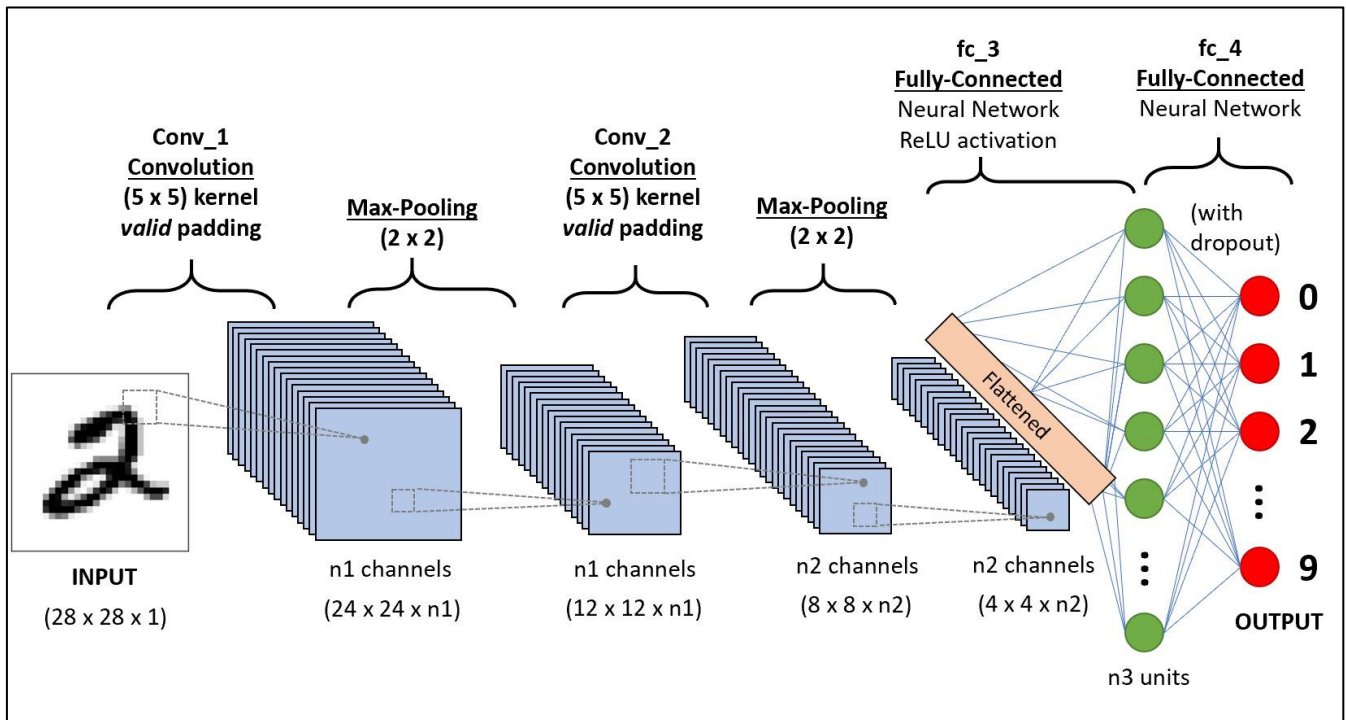


Рисунок 2.3 — Загальна архітектура згорткової нейронної мережі

Об'єднання у пул використовується для того, щоб отримавши функції за допомогою згортки, їх можна було використати для класифікації. Теоретично можна було використати всі витягнуті функції з за допомогою деякого класифікатора, наприклад ReLU, але це може бути складно. Розглянемо, наприклад, зображення розміром 96×96 пікселів, і припустимо, що отримано 400 функцій на 8×8 входах. Кожна згортка призводить до виходу розміру $(96 - 8 + 1) * (96 - 8 + 1) = 7921$, і оскільки є 400 особливостей, це призводить до вектору $7921 * 400 = 3168400$ ознак.

Вивчення класифікатора з вхідними даними, що мають майже 3 мільйони функцій, може бути непростим, а також потребувати великої кількості обчислювальних ресурсів.

На наступному зображенні (рисунок 2.4) [10] показано, як об'єднання виконується для 4 не перекритих областей зображення. В результаті для великої кількості даних утворюється набагато менша матриця, яку легше обробляти і цей процес потребуватиме менше ресурсів.

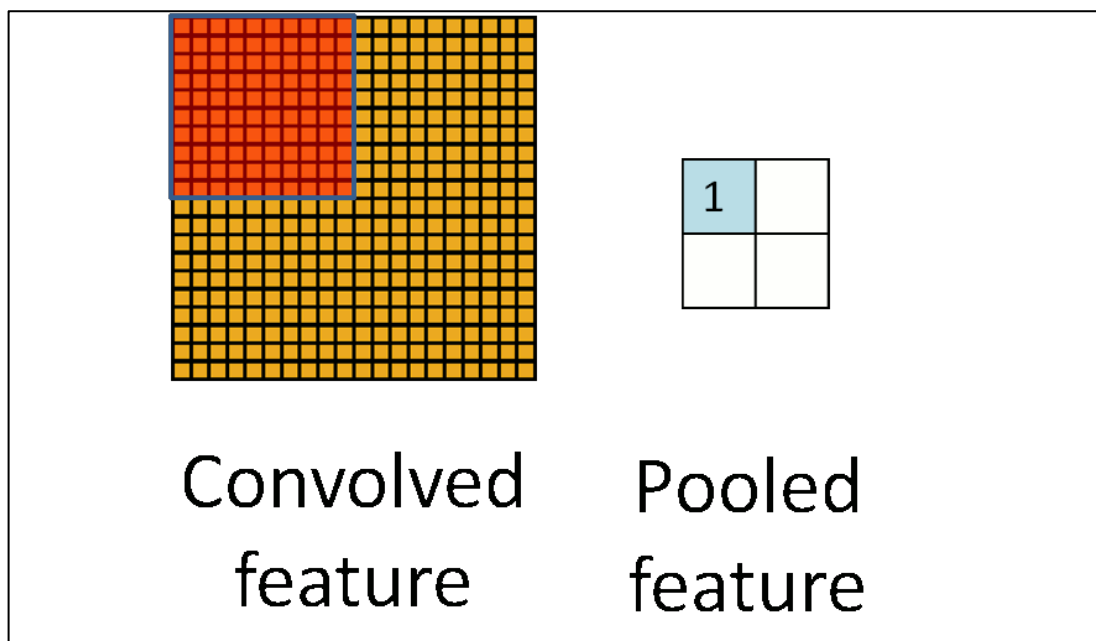


Рисунок 2.4 — Об'єднання згорткових характеристик в пул

Для вирішення цього питання вирішено отримати переплетені функції, оскільки зображення мають властивість «стаціонарності», що означає, що функції, корисні в одному регіоні, також можуть бути корисними для інших регіонів. Таким чином, для опису великого зображення, одним із природних підходів є узагальнення статистики цих особливостей у різних місцях. Наприклад, можна обчислити середнє (або максимальне) значення певної особливості для області зображення. Ця підсумкова характеристика значно менша за розмірами (порівняно з використанням усіх функцій), а також може покращити результати. Операцію агрегації називають цією операцією "об'єднання", а іноді "середнє об'єднання" або "максимальне об'єднання".

Останнім часом найпопулярнішими нейронними мережами вважаються саме CNN. Їх використовують найчастіше для аналізу зображень. У роботах [11], [12], [13] та [14] вирішується задача розпізнавання звучання музичних інструментів. В першій роботі пропонується розв'язати задачу розпізнавання для 10 інструментів і для порівняння обрати метод SVM та CNN. Архітектура CNN покладається на те що спочатку з аудіо-сигналу буде виділено спектрограму і аналіз буде відбуватись для отриманого зображення. В якості характеристик пропонується обрати MFCC. Отримані результати виявились досить поганими і показали в середньому 50%

правильних відповідей. В другій роботі дослідження проводиться на тренувальній вибірці IRMAS. [15] Для порівняння було обрано три набори характеристик STFT, MFCC та HAS-IMF. Найкращі результати показали характеристики MFCC, в роботі досягається результат у 77% правильних класифікацій. В роботі також пропонується використовувати архітектуру CNN, яка на вході очікуватиме зображення спектрограми.

В третій роботі порівнюються архітектури двох нейронних мереж, а саме RNN та CNN для вирішення задачі розпізнавання звучання музичних інструментів. В обох випадках на вхід подається спектрограма, тобто зображення музичних характеристик. В досліді використовуються перетворення Фур'є з фільтрацією у 64 та 128 діапазонів, що охоплюють від 0 до 22050Гц, після чого обчислюється значення децибел відносно пікової потужності, і дані нормалізуються. В результаті експерименту, було зроблено висновок, що CNN навчаються швидше та показують кращі результати розпізнавання ніж RNN для задачі розпізнавання музичних інструментів. В загальному CNN показала результат у 83%, а RNN – 81.9%.

В останній роботі використовується дуже велика навчальна вибірка, в 1 мільйон записів. Для аналізу використовуються характеристики гармонічних серій та константне Q-перетворення. В результаті було розроблено класифікатор, який показав дуже хороший результат – 93% в деяких випадках. Однак, для деяких інструментів цей результат був дуже низьким, наприклад для валторни результат 81.3%, в той час як піаніно показує результат у 99,7%. Однак, варто враховувати, що розмір навчальної вибірки дуже великий, що не могло не відобразитись на результатах.

В результаті аналізу попередніх досліджень вирішення задачі класифікації звучання музичних інструментів за допомогою нейромережових методів можна прийти до висновку, що MLP дуже прості нейронні мережі і з них варто розпочати дослідження даної сфери, оскільки вони мають просту архітектуру і легкі в вивченні. Вони можуть давати хороші результати на рівні класичних методів розв'язання задачі. Однак, найбільшу цінність для вивчення мають згорткові нейронні мережі, які показують кращі результати ніж багат шарові та рекурентні нейронні мережі.

Згорткові нейронні мережі складніші для розуміння, однак показують набагато вищу точність.

2.4 Засоби візуалізації навчальної вибірки

Розмірність – це важлива тема в машинному навчанні та науці даних, яка описує потенційну складність набору даних. Набір даних містить безліч точок даних, кожна з яких має постійну кількість функцій або розмірів. Наприклад, якщо розглядати звучання музичних інструментів, то кількість обраних характеристик буде вказувати на розмірність даних.

Зрозуміло, що зі збільшенням розмірності, пошук розв'язку вимагає більших витрат часу та ресурсів. Це справедливо як для людей, так і для комп'ютерів. Ще одне важливе питання полягає в тому, що потрібно більше даних для точного моделювання простору з більшими розмірами; зі збільшенням розмірності об'єм простору зростає експоненціально.

Зменшення розмірності є важливою темою в практиці машинного навчання, оскільки висока розмірність може призвести до високих обчислювальних витрат, а також до тенденції до надмірного використання даних.

Зменшуючи розмірність, зменшуються розміри наборів даних. Чим більша кількість розмірів, тим складніше її візуалізувати.

Найбільш простим підходом до зменшення розмірності може бути вибір підмножини функцій, що найкраще описує дані, та відкидання решти вимірів, що називається вибором характеристик (feature selection). При цьому втрачається деяка інформація.

Кращим рішенням є приведення даних до меншої розмірності. Цей підхід називається вилученням характеристик (feature extraction).

Метод головних компонент – це один зі способів зменшити розмірність даних втративши найменшу кількість інформації. Перетворення відбувається на основі одного з чотирьох методів:

- 1) апроксимація даних лінійними перетвореннями меншої розмірності;

2) пошук простору меншої розмірності, в ортогональній проекції на яку розкид даних максимальний;

3) пошук простору меншої розмірності, в ортогональній проекції на яке середньоквадратична відстань між точками максимальна;

4) побудова ортогонального перетворення координат, в результаті якого кореляція між окремими координатами зведеться до 0.

Цей метод широко розповсюджений для візуалізації даних, і в даному дослідженні використовується для візуалізації характеристик музичних інструментів. Також він використовується для приглушення шуму на зображеннях, індексації відео, біо-інформатиці, політології, соціології, економетриці та психодіагностиці.

Наступний алгоритм зменшення розмірності, t-розподілене вкладення стохастичної близькості (t-SNE), був розроблений для наборів даних високих розмірів Лоренсом ван дер Маатеном та Джеффри Хінтоном.

У алгоритму t-SNE є два етапи. Спочатку будується розподіл ймовірності за парами об'єктів високого розміру, так що більш подібні об'єкти будуть обрані з більшою ймовірністю. Потім будується аналогічний розподіл ймовірностей для малих розмірних карт. Розбіжність між двома розподілами ймовірностей потім мінімізується. Ця розбіжність, або відносна ентропія, називається дивергенцією Куллбека – Лейблера. [17]

Перший аргумент алгоритму – це заплутаність (perplexity), яка стосується кількості найближчих сусідів, що використовуються багатоманітних алгоритмах навчання. Інший параметр – сума ітерації, яка полягає в тому, як довго оптимізувати t-SNE. Кількість ітерацій має набагато більший вплив на отримані результати ніж заплутаність. В офіційній документації вказується, що продуктивність роботи алгоритму є досить надійною щодо змін у заплутаності, а типові значення – від 5 до 50. В даному дослідженні алгоритм t-SNE використовується для візуалізації отриманого тренувального набору даних. Роботу даних алгоритмів проілюстровано в розділі програмної реалізації продукту.

Висновки до розділу 2

Цей розділ було присвячено аналізу попередніх підходів до вирішення задачі класифікації звучання музичних інструментів. В ході було розглянуто класичні та сучасні методи.

В результаті аналізу було встановлено, що незважаючи на відмінності та особливості проведення експериментів, всі вони показують, що доволі складно однозначно визначити найкращу комбінацію вибору характеристик та методів розпізнавання. Найкращі результати показують дослідження, в яких використовуються згорткові нейронні мережі, однак точність в даних дослідженнях коливається від 50% до 93%, що підтверджує тезу про те що остаточна точність будь-якого класифікатора явно залежить від вибраної кількості та типу включених інструментів, а також від характеристик та підготовки прикладів для кожного інструмента.

В кінці розділу було розглянуто основні методи візуалізації тренувальної вибірки, що дозволяє провести якісну оцінку обраних даних. До таких методів належать PCA та t-SNE.

3 ХАРАКТЕРИСТИКИ ЗВУЧАННЯ

В цьому розділі представлено огляд основних характеристик аудіосигналів, які використовуються в прикладних задачах класифікації. Звуковий сигнал характеризується спектральними, часовими, енергетичними та ймовірнісними характеристиками. Перші два типи найчастіше використовуються на практиці. Часові характеристики визначають, які з аспектів звукового сигналу змінюються протягом звучання ноти. До часових характеристик належать: частота перетину нуля (zero-crossing rate), пік-фактор (amplitude factor), кількість ударів за хвилину (beats per minute), середньоквадратичне значення енергії (root mean square energy). До спектральних характеристик належать: мел-частотні кепстральні характеристики (MFCCs), спектральний центроїд (spectral centroid), спектральний коефіцієнт рівнинності (spectral flatness).

3.1 Часові характеристики

Опис сигналу деякої функції часу повністю визначає її властивості. При часовому описі коливань, значення виражається як функція від часу або як сума більш простих часових функцій. Гармонічний ряд визначається наступною формулою (3.1) [18]:

$$U(t) = U_0 + U_m \cos(\omega t + \varphi_0), \quad (3.1)$$

де U_0 – постійна складова, U_m – змінна складова амплітуди коливання, φ_0 – початкова фаза, $\omega = \frac{2\pi}{T}$ – циклічна частота.

Приклад часового представлення даного ряду (рисунок 3.1) надано для наступних значень характеристик:

$$U_0 = 2\text{В}, U_m = 5\text{В}, T = 0,8; f(x) = \frac{1}{T} = 5.$$

Таким чином, $U(t) = 2 + 5 \cos(1,25 \cdot 2\pi t + 0)$.

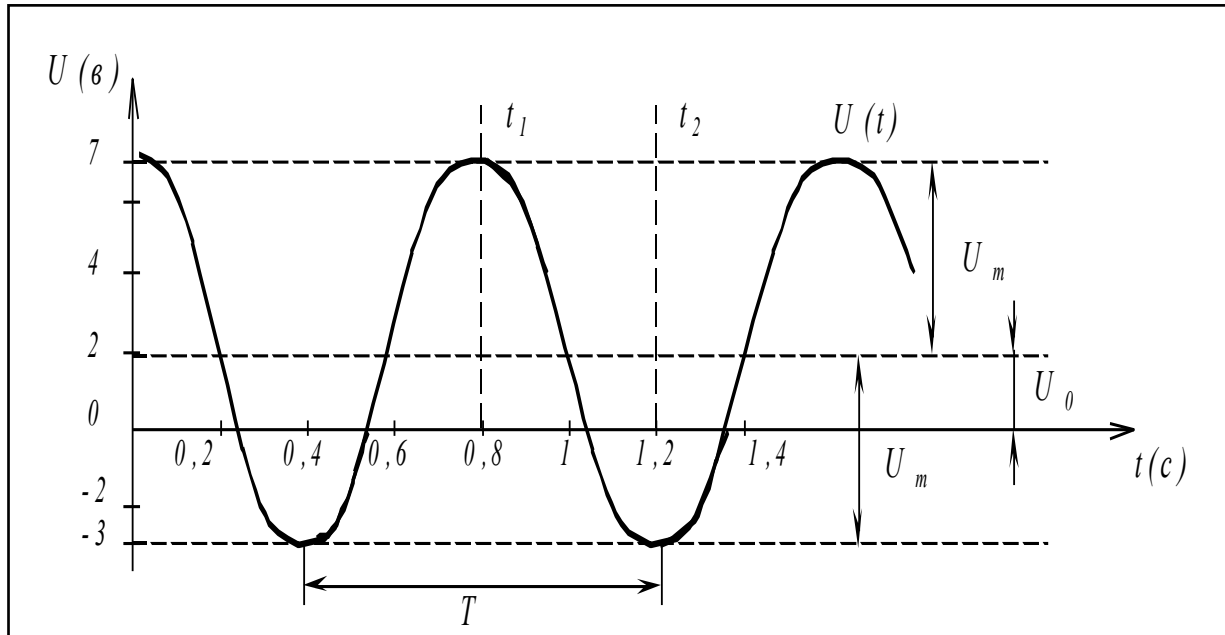


Рисунок 3.1 – Часове представлення гармонічного коливання

Нижче наведено базові часові характеристики, які можуть бути використані або вже використовувалися в попередніх дослідженнях.

Частота перетину нуля (zero-crossing rate) – це частота зміни знаків вздовж сигналу, тобто частота, з якою сигнал змінюється від позитивного до негативного і навпаки. Характеристика застосовується як для класифікації музичних інструментів, так і в задачах розпізнання мовця, розраховується за формулою:

$$Z_t = \frac{1}{2} \sum_{n=1}^N |\text{sign}(x[n]) - \text{sign}(x[n-1])|,$$

де функція sign приймає значення 1, якщо аргумент додатний, 0, якщо аргумент від'ємний, $x[i]$ – значення i -го елементу. [18]

Однак, так як в даному дослідженні передбачається безпосереднє використання спектральних характеристик, надалі частота перетину нуля не розглядається в якості значимої характеристики аудіосигналу.

Пік-фактор – це відношення пікової потужності до середньої, вираховується з формули:

$$PF = \frac{P_{max}}{P_{av}}$$

Пік-фактор може бути застосований для відокремлення піцикато (pizzicato) від стійкого (sustain) тону. Піцикато розсіюється експотенційно та має вище значення пік-фактору. Однак, репозиторії прикладів музичних інструментів зазвичай не надають інформацію щодо використання різноманітних музичних технік в представлених записах.

Середньоквадратичне значення енергії (Root mean square (RMS) energy) визначає ємність амплітуди звукової хвилі водному аналітичному вікні, значення RMS можна вирахувати з формули:

$$RMS = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}},$$

де n – кількість зразків, x – енергія зразка. [18]

Нижче наведено приклади значень даної характеристики для тромбона та скрипки (рисунок 3.2), побудовані з використанням бібліотеки обробки звукових записів Librosa:

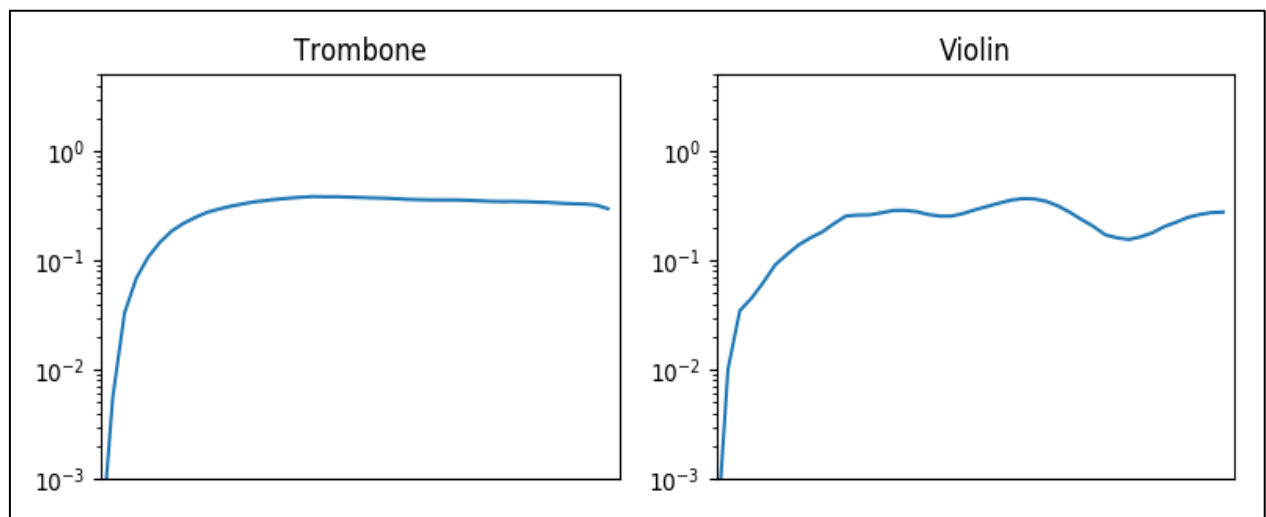


Рисунок 3.2 – Характеристика середньої енергії для тромбону та скрипки

Фактично RMS – це міра енергії присутньої в заданій ноті. Згідно з [11] дана характеристика є визначальною для тону гри музичного інструменту. Як видно з графіку, доволі складно відрізнити звучання інструментів за цією характеристикою, тому вона не враховується під час дослідження.

3.2 Спектральні характеристики

Спектр сигналу характеризує розподіл енергії сигналу за частотами. Властивості спектрів необхідно враховувати для забезпечення не викривленої передачі інформації. Для отримання спектрів необхідно вміти описувати або представляти їх у вигляді ряду Фур'є.

При спектральному описі значення представляють у вигляді суми складових різних частот. Відповідно спектральне представлення формули (3.1) матиме вигляд графіку приведенного нижче (рисунок 3.3). [21]

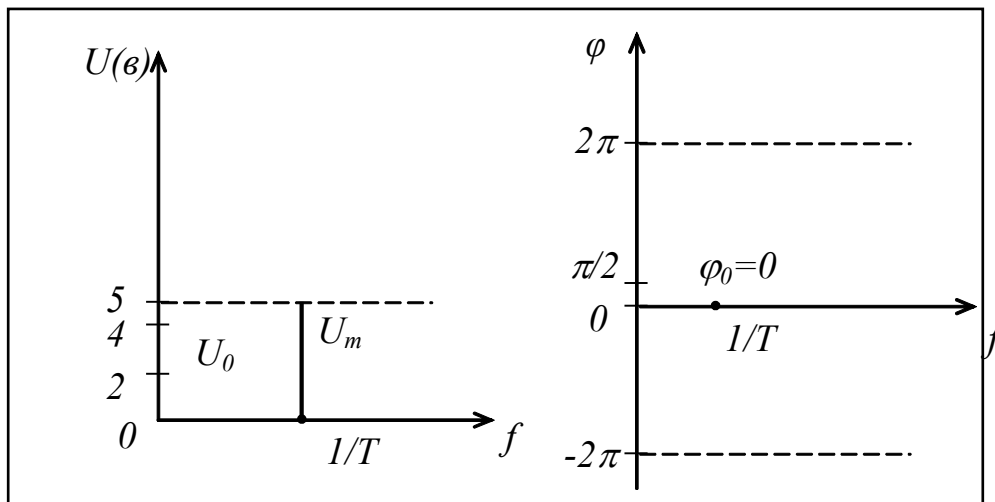


Рисунок 3.3 – Спектральне представлення гармонічного коливання

Нижче розглянуто спектральні характеристики звукового сигналу, що знайшли широке застосування в вирішенні задачі класифікації музичних інструментів.

Мел-частотні кепстральні коефіцієнти (Mel-frequency Cepstral Coefficients, MFCCs) – це набір коефіцієнтів, які дозволяють представити спектральні якості звуку з масштабуванням, оснований на слуху людини.

Отримання даних характеристик передбачає наступні кроки:

1) розподілити сигнал на фрейми. Це зазвичай відбувається за допомогою віконного перетворення до звукових сигналів на фіксованих інтервалах;

2) отримати амплітуду спектру для кожного фрейму шляхом застосування дискретного перетворення Фур'є. В подальших розрахунках використовується лише логарифм амплітуди спектру, тоді як фаза вважається не значимою характеристикою;

3) перетворення сигналу покепстрово на мели. Частота f перетворюється на мели m відповідно до формули:

$$m = 1127.01048 * \ln \ln (1 + f/700)$$

4) застосувати дискретне косинус перетворення до компонент мел-векторів для зменшення кореляції між ними. [22]

На рисунку 3.4 наведено приклади кепстральних коефіцієнтів для тромбону та скрипки.

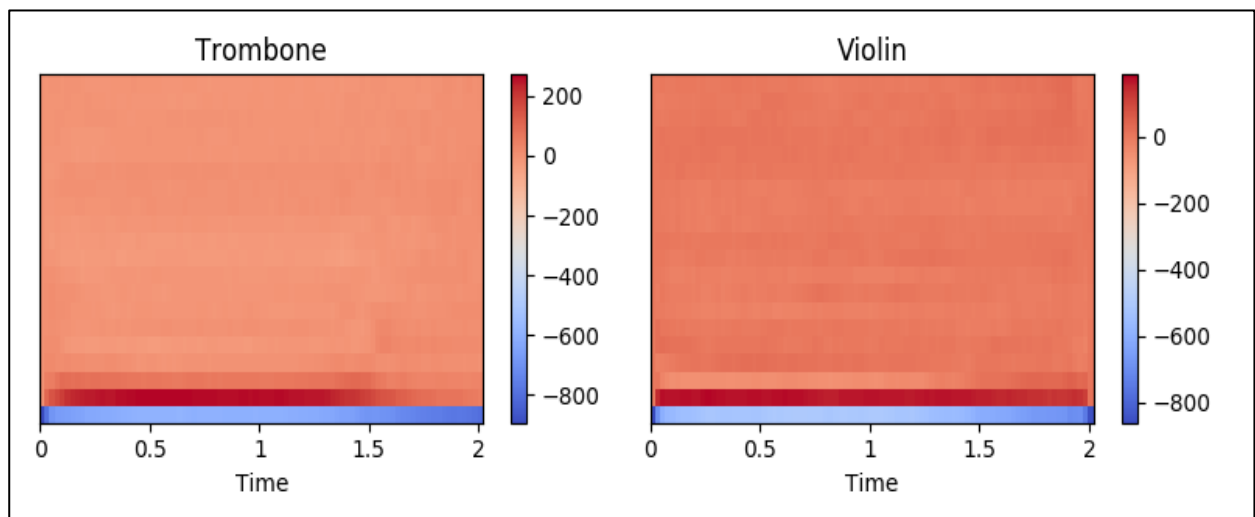


Рисунок 3.4 – Мел-кепстральні характеристики тромбону та скрипки

Мел-кепстральні характеристики є одними з найважливіших характеристик, і використовуються в багатьох дослідженнях розпізнавання та класифікації аудіосигналів.

Спектральна рівнинність (Spectral flatness) виражає відхилення потужності спектра сигналу від пологої форми. З точки зору людського сприйняття, характеризує ступінь тональності звукового сигналу. Розраховується як відношення середньо-геометричного значення спектральної щільності до її середньоарифметичного значення.

Розрахувати спектральну рівнинність можна за формулою:

$$SF = \frac{\sqrt[N]{\prod_{n=0}^{N-1} x(n)}}{\frac{\prod_{n=0}^{N-1} x(n)}{N}},$$

де $x(n)$ – значення магнітуди для інтервалу n .

На рисунку (рисунок 3.5) представлено приклади значень даної характеристики для двох музичних інструментів.

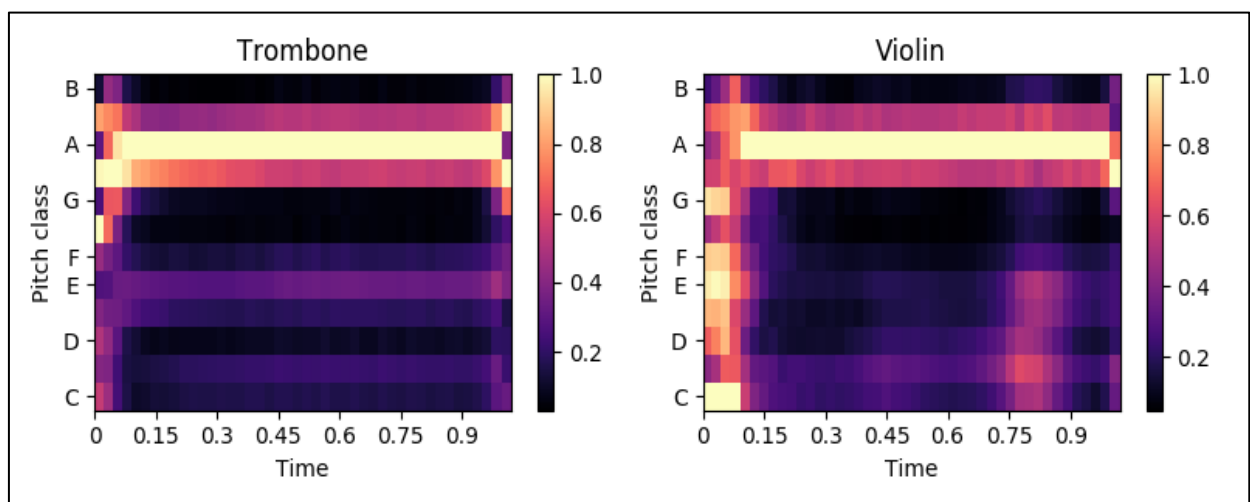


Рисунок 3.5 – Характеристика тональності для тромбону та скрипки

Як і перетворення Фур'є, постійне перетворення Q є набором ітерацій, але на відміну від першого воно має геометричний розмір центральних частот.

Спектральний центроїд є мірою, що використовується для обробки цифрових сигналів для характеристики спектра. Він вказує, де знаходиться "центр мас" спектра. Він має сильний зв'язок із вираженням "яскравості" звуку. Порівняння характеристики для тромбону та скрипки представлені нижче (рисунок 3.6).

Спектральний коефіцієнт рівнинності або тонічності (spectral flatness), є мірою, що використовується для обробки цифрових сигналів для характеристики спектру звуку. Спектральна плоскості, як правило, вимірюється в децибелах і дає змогу кількісно визначити, наскільки звук шумовий, на відміну від тонального.

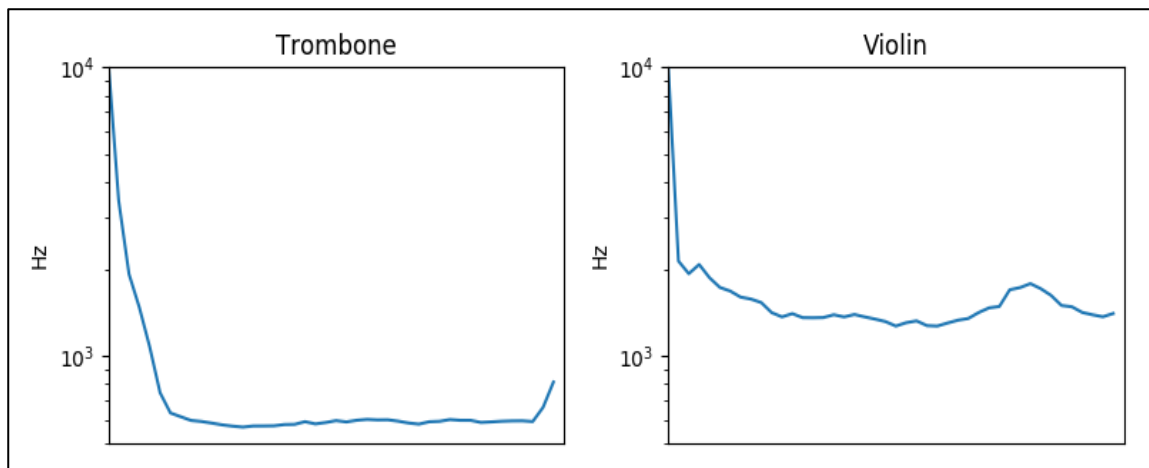


Рисунок 3.6 – Характеристика спектрального центроїду для тромбону та скрипки

Частота згортання. Згортка (roll-off) – це крутизна передавальної функції за частотою. Її можна розрахувати за наступною формулою:

$$\text{MIN}(R) \text{ where } \sum_i^R \text{mag}[i] \geq 0.85 \times \sum_{i=0}^N \text{mag}[i]$$

де N – довжина сигналу, $\text{mag}[i]$ – магнітуда інтервалу i . [19]

Приклади даної характеристики наведено на рисунку 3.7:

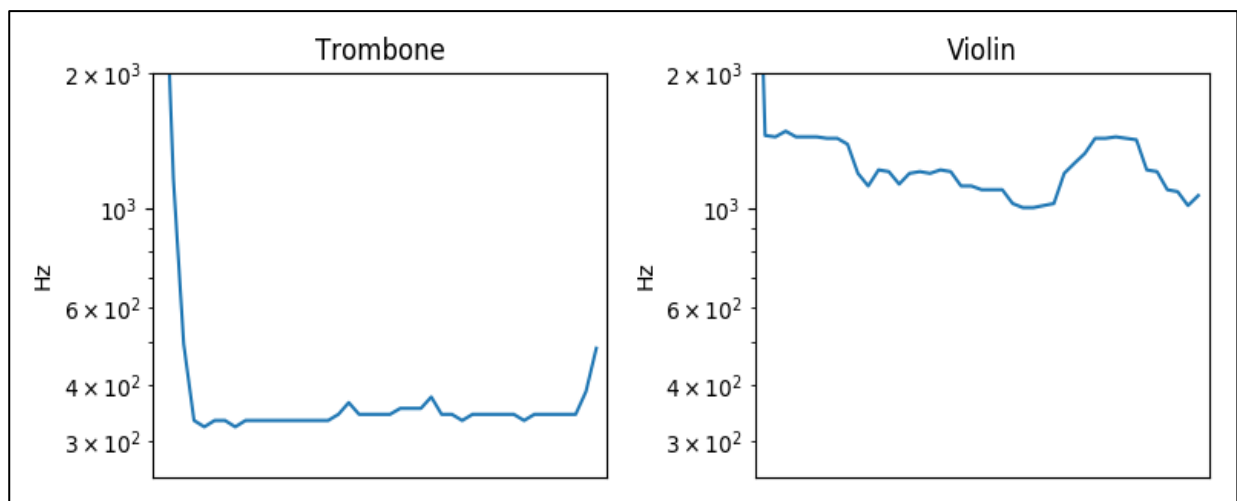


Рисунок 3.7 – Характеристика згортки для тромбону та скрипки

Спектральна частота згортання є мірою розподілу енергії за звуком. Зокрема вона характеризує частоту, нижче якої в звуці міститься певний рівень енергії (зазвичай обирають 85%).

3.3 Обрані характеристики для дослідження

Проаналізувавши попередні дослідження у другому розділі та виконавши огляд характеристик звучання музичних інструментів можна приступити до етапу формування навчальної вибірки. У роботі [16] було проведено дослід з використанням мел-кепстральних характеристик для розпізнавання звучання музичного інструменту і було зроблено висновок, що окрім них для покращення результату можна використовувати їх в комбінації разом з іншими часовими та спектральними характеристиками. [23]

В результаті аналізу музичних характеристик та їхнього впливу на якість розпізнавання, а також результатів попередніх робіт для даного дослідження було обрано наступні характеристики:

- спектральний центроїд;
- мел-частотні кепстральні характеристики.

Висновки до розділу 3

У розділі наведено огляд важливих характеристик аудіосигналів. Виокремлюють чотири типи музичних характеристик, однак на практиці широкого застосування набули часові та спектральні. В розділі представлено їх математичний зміст. Наведено приклади значення даних характеристик для двох класів музичних інструментів – духових та струнних, а саме тромбону та скрипки.

Для подальшого тренування класифікатору обрано наступні характеристики: спектральний центроїд та мел-частотні кепстральні характеристики.

4 ЗАСОБИ РОЗРОБКИ

Машинне навчання – це область з величезним вибором інструментів, бібліотек та фреймворків. Однак велика кількість варіантів призводить до розгубленості.

Вибір фреймворку в перспективі може стати або перевагою, або недоліком майбутнього проекту. Фреймворк диктує правила побудови архітектури програмного забезпечення. Ще на початковому етапі розробки задається поведінка за замовчуванням, каркас, який необхідно розширювати та змінювати відповідно до вказаних вимог протягом усього процесу розробки, а зміна архітектури проекту дуже затратна процедура. З іншого боку використання певного фреймворку дозволяє легко супроводжувати, модернізувати та масштабувати проект, відносно просто реалізовувати будь-які бізнес процеси. Рішення на фреймворках, як правило, працюють значно швидше, безпечніші і витримують більші навантаження, ніж самописні системи.

Отже, вибір інструментів розробки є важливим етапом створення програмного забезпечення, і в цьому розділі розглядатимуться основні підходи та програмні засоби, їхні плюси та мінуси для розробки продуктів з використанням машинного навчання.

4.1 Бібліотеки музичних записів

Характеристики навчальної вибірки мають безпосередній вплив на ефективність навчання НМ. З метою формування навчальної вибірки було проведено аналіз найпоширеніших репозиторіїв зразків музичних записів (таблиця 4.1). Пошук репозиторіїв відбувався за допомогою аналізу попередніх досліджень, а також під час пошуку баз даних, що зберігають приклади звучання гри музичних інструментів, наприклад сайти філармоній та оркестрів.

В дослідженні було розглянуто як комерційні репозиторії так і ті, що поширюються за вільною ліцензією. Для аналізу було обрано такі характеристики як типи представлених музичних інструментів та формати даних

Таблиця 4.1. Репозиторії зразків музичних записів

Назва	Опис	Розробник	Типи музичних інструментів	Вільна ліцензія
McGill University Master Samples	Бібліотека прикладів, представлених в WAV форматі	Opolko & Warnick	Струнні, клавішні, духові, ударні	Так
Musical instrument sample	Бібліотека музичних інструментів Університету Айови, представлених у форматі AIFF	Fritts	Струнні, клавішні (фортеп'яно), духові, ударні	Так
Vienna Symphonic Library	Бібліотека віртуальні інструментів та відтворення акустик знаменитих концертів	Vienna Symphonic Library GmbH	Струнні, арфи, духові, хори, саксофон, трубний орган,	Ні
London Philharmonic Orchestra dataset	Записи у виконанні оркестру філармонії, представлених у форматі mp3	London Philharmonic Orchestra	Струнні, клавішні, духові, ударні	Так
IRMAS	Містить уривки з анотаціями домінуючих інструментів, 9579 аудіо-файлів в wav форматі	Universitat Pompeu Fabra	Флейта, гітари, орган, клавішні, кларнет, скрипка	Так
Free sound	Бібліотека музичних записів, представлених у різних форматах	Free sound community	Струнні, клавішні, духові, ударні	Так

Серед представлених репозиторіїв історично найбільш широкого застосування в академічній сфері набула бібліотека MUMS. Бібліотека складається з трьох DVD-дисків та містить зразки більшості стандартних класичних, деяких нестандартних класичних та багатьох популярних музичних інструментів. Наразі у бібліотеці 6546 зразків записів, розділених між струнними (2204), клавішними (1595), перкусією (1087), дерев'яними (1197) та мідними духовими (463) сімействами музичних інструментів. Кожна нота з кожного інструменту була записана окремо (44,1 кГц, 24 біт). Для більшості інструментів представлено декілька стилів артикуляції. Як правило, для кожного приладу є 29 зразків, що означає, що охоплюється не весь діапазон висот інструментів. Тривалість зразків переважно становить від 2 до 10 секунд та може значно відрізнятися в межах декількох інструментів. Це може зумовлювати проблеми при використанні темпоральних характеристик, оскільки вони безпосередньо залежать від часу. Бібліотека містить ряд помилок, які підлягають виправленню у нових ревізіях. Крім того, існує ряд труднощів в процедурі доступу до репозиторію, адже його необхідно замовляти на сайті університету з обґунтуванням причин використання.

Віденська симфонічна бібліотека (Vienna Symphonic Library) надає віртуальні інструменти та цифрове відтворення акустики знаменитих концертних залів. З моменту створення було записано та опубліковано більше трьох мільйонів зразків практично всіх інструментів симфонічного оркестру, хору та інших. Це більш терабайт звуку з високою роздільною здатністю та найбільша база зразків, яку коли-небудь створювала одна компанія. В той же час, записи поширюються на комерційній основі, що обмежує можливості їх використання в навчальних цілях.

Бібліотека Musical instrument sample (MIS) була створена в університеті Айови в 1997 році і з того часу доступна для вільного завантаження на його сайті. [28] Записи відбувалися для кожного інструменту нота за нотою на спеціалізованому обладнанні. На разі в репозиторії представлено 23 музичних інструменти. Бібліотеку поділено на два підрозділи відповідно до технології запису зразків: до 2012 та після 2012 року. Недоліком бібліотеки є те, що кожен інструмент необхідно завантажувати окремо та відсутності опису характеристик записів.

Репозиторій Freesound має на меті створити величезну колективну базу даних аудіо фрагментів, зразків, записів, добірок. Наразі користувачі завантажили тисячі записів, які поширюються за вільною ліцензією. Однак створення навчальної вибірки з такого репозиторію потребує кропіткої праці, щодо екстракції необхідних записів та перевірки їх на відповідність вимогам щодо якості.

Репозиторій IRMAS було створено для використання для тренування та тестування методів для автоматичного розпізнавання домінантного інструменту. Уривки представлені для багатьох музичних інструментів в загальному числі в 9579 записів на частоті 44.1кГц. Звучання інструменту представлено тривалістю в 3 секунди.

На основі порівняння архівів зразків записів музичних інструментів навчальної вибірки для подальшого вирішення задачі класифікації було обрано бібліотеку Лондонського симфонічного оркестру. Вона містить у собі тисячі екземплярів мелодій, які поширюються за вільною ліцензією, та широко застосовується для наукових досліджень в області розпізнавання звукових сигналів. В бібліотеці представлено приклади 20 різних інструментів з 4 різних класів – струнних, духових, ударних та клавішних. Зразки охоплюють всі набори тонів в кожній октаві та різній силі звучання (форте та піано). Крім того набори містять зразки різної техніки гри, такі як, наприклад, вібрато та тремоло. Крім того, репозиторій має зручну структуру представлення записів. Користувач має змогу завантажити всі представлені дані або обрати музичний інструмент чи перкусію. В рамках одного музичного інструмента надається змога сортування та фільтрації за обраними характеристиками, такими як: висота звуку, довжина запису, динаміка та артикуляція. Представлена можливість завантажити всі зразки для даного інструменту або обрати лише декілька відповідно до необхідних характеристик.

Для збільшення об'ємів даних можна поєднувати дані з різних репозиторіїв, однак можливо для цього потрібно буде розробити додатковий модуль, який буде виконувати екстракцію музичних характеристик. Для прикладу можна обрати бібліотеки Лондонського симфонічного оркестру та IRMAS, однак дані у них

представлені в різних форматах, тому спершу потрібно буде провести екстракцію даних в форматі mp3 та wav, а потім поєднувати отримані дані в загальне сховище.

В даному дослідженні було обрано наступні струнні та духові інструменти:

- фагот;
- скрипка;
- контрабас;
- тромбон;
- туба;
- флейта;
- віолончель;
- валторна.

В початкову вибірку для кожного з інструментів виокремлено понад 900 зразків з тривалістю запису 1 секунду. Усього було отримано 7294 зразки, 1500 з яких виокремлено для тестової вибірки, а решта 5794 – для тренувальної.

4.2 Засоби екстракції музичних характеристик

Для обробки музичних записів та екстракції характеристик було розглянуто 3 найпоширеніші бібліотеки – Aubio, Yaafe та LibROSA.

Aubio – це утиліта для екстракції анотацій з аудіосигналів. Її функції включають сегментацію звукового сигналу перед кожною його атакою, виявлення тембру, удару та виробництво потоків з онлайн аудіо.

В даний час бібліотека надає наступні функції:

- цифрові фільтри;
- фазовий волокодер;
- початок виявлення;
- відстеження тембрів;
- відстеження темпу та біту;
- мел-кепстральні коефіцієнти частоти.

Бібліотека написана на мові C і підтримує інтерфейс мови Python. [33]

Yaafe – це набір інструментів для обробки аудіозаписів. Вона підтримує зчитування популярних аудіо форматів, збереження вихідних даних у CSV та HDF5 формати, має інтерфейс з мовою Python та великий перелік класичних аудіо характеристик, таких як мел-кепстральні коефіцієнти частоти, енергію, магнітудний спектр, спектральну рівнинність та багато інших. [34]

Однак важливим недоліком є брак прикладів використання бібліотеки, що призвело до того що з нею досить складно почати працювати.

LibROSA – бібліотека мови Python для аналізу музики та аудіо. Вона міститься в пакетному менеджері PIP і тому її досить швидко та легко встановити.

Бібліотека забезпечує екстракцію основних характеристик аудіосигналів, а також має чудову документацію та безліч прикладів щодо використання її функцій.

4.3 Засоби розробки нейронних мереж

Існує декілька факторів, які суттєво впливають на вибір фреймворку. Всі вони необхідні по-своєму і ступінь важливості кожного з них залежить від прикладної задачі. Деякі фреймворки є лідерами в певних аспектах, але погано задовольняють іншим вимогам. Існують і такі, що є поєднанням усіх факторів на задовільному рівні.

Переглянемо основні аспекти при виборі фреймворку:

— доступність навчальних матеріалів. Важливий, але зазвичай недооцінений аспект. Деякі бібліотеки та фреймворки можуть бути представлені простими та елегантними елементарними прикладами, однак під час розробки виникають ситуації коли потрібно більш детально заглибитись в можливості засобу розробки і, окрім сухої документації, знадобляться додаткові курси, книги та статті. Якісні навчальні ресурси значно скоротять процес навчання;

— популярність. Вивчення екзотичної технології може бути цікавим процесом, однак на практиці краще надавати перевагу «перевіреному у бою» набору інструментів. Цьому слугує декілька причин. Якщо фреймворк не дуже популярний, це означає, що є не так багато розробників, спеціалістів у ньому. Другою причиною

є відсутність спільноти, до якої можна звернутись за допомогою і залишившись один на один з документацією, з великою ймовірністю буде згаяно багато часу;

— основні особливості та можливості інструменту. Цей фактор напряму вказує на те, як і що ви зможете робити з обраним фреймворком, які обмеження він накладатиме на програмне забезпечення;

— простота використання. Найкращим способом зрозуміти як працює фреймворк – використати його простому проєкті. Це дасть можливість зрозуміти чи дозволяє фреймворк вирішувати повсякденні проблеми перераховані вище. При роботі з фреймворком слід також врахувати продуктивність розробки, а також рівень залежності фреймворку від інших інструментів. Варто звернути увагу на доступні засоби програмного забезпечення, такі як IDE та розширення;

— швидка інтеграція з іншими сервісами. Цей аспект тісно пов'язаний з попереднім. Не важливо наскільки функціонально багатий обраний фреймворк, все одно є шанси наštтовхнутись на проблему, для вирішення якої знадобляться додаткові інструменти. Є безліч бібліотек, які фокусуються на вирішенні окремих задач. Якщо при спробі підключити одну кожного разу буде витрачатись декілька годин – це, мабуть, не оптимальний варіант.

Нижче (таблиця 4.2) представлено найпоширеніші на сьогодні фреймворки машинного навчання.

Таблиця 4.2. Фреймворки для роботи з машинним навчанням

Фреймворк	Розробник	Мови інтерфейсу	Посилання на офіційний сайт
Caffe2	Facebook	Python, C++	[25]
PyTorch	Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan	Python	[26]
CNTK	Microsoft	Python, C#	[27]
MXNet	Apache Software Foundation	Python, C++, R, Julia, Perl, Scala	[28]
TensorFlow	Google Brain team	Python, C/C++, Java, Go, R, Julia	[29]
Apache SINGA	Apache Incubator	Python, C++, Java	[30]

Caffe – середовище глибинного навчання, розроблене Яцзином Цзя (Yangqing Jia) в університеті Берклі. Є відкритим програмним забезпеченням, що розповсюджується під ліцензією BSD license. Написана на мові C++, і підтримує інтерфейс на мові Python. Фреймворк завдяки своїй продуктивності чудово підходить для роботи з зображеннями, має велику кількість функцій для їхньої обробки.

В грудні 2017 Facebook оголосив про створення Caffe2, що включила наступні можливості:

- мобільну розробку;
- нову апаратну підтримку;
- підтримку широкомасштабного розподіленого навчання;
- гнучкість для майбутніх напрямів, таких як квантове обчислення;
- стрес тестування великою кількістю застосунків Facebook.

Перевагою Caffe2 є також можливість інтегрувати бібліотеку з іншими сервісами. Наприклад, AWS додали підтримку Caffe2 в Deep Learning AMI, Qualcomm працює над оптимізацією Snapdragon для роботи з Caffe2, про підтримку фреймворку також оголосили Intel та NVIDIA.

Основною архітектурною відмінністю Caffe та Caffe2 стала зміна основного об'єкту. У Caffe ним був «шар», а тепер використовується «оператор» (рисунк 4.1). [25] Це допомогло вирішити основну проблему Caffe – неможливість побудувати інтерактивний «граф» мережі.

Caffe2 має свої недоліки. Він несумісний зі старою версією фреймворку, однак варто зауважити, що цей недолік не є суттєвим для нових розробок. Об'єм коду збільшився, підвищилась складність програм, погано опрацьована структура проекту. Також, важливим недоліком є відсутність достатньої кількості навчальних матеріалів, курсів та книг. Невелика кількість прикладів використання ускладнює початок взаємодії з фреймворком. Для того щоб використати ту чи іншу функцію доводиться багато часу проводити за читанням документації, яка покращилась, але іноді її буває недостатньо і через це продуктивність розробки на Caffe2 не висока.

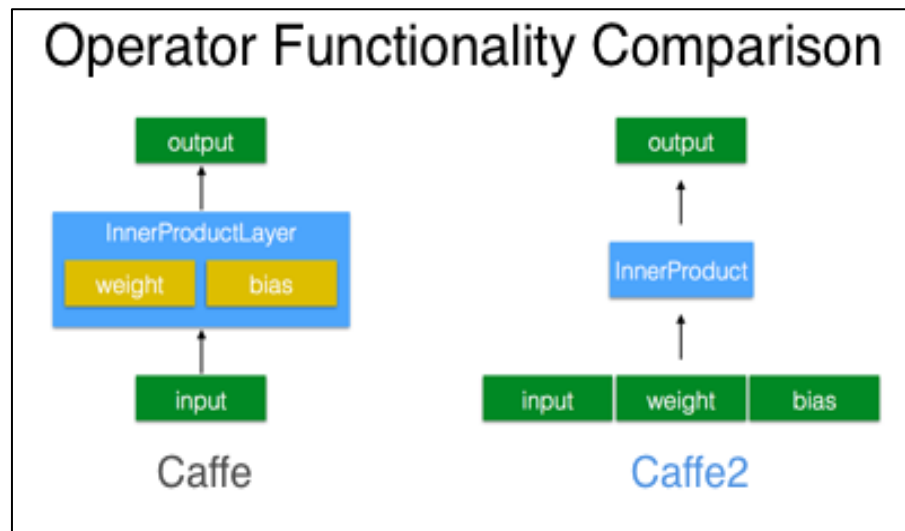


Рисунок 4.1 – Порівняння функціональності операторів Caffe та Caffe2

Загальний висновок – Caffe2 хороший сучасний фреймворк для роботи з машинним навчанням, з активною спільнотою, хорошою продуктивністю, яка дає можливість обробляти велику кількість зображень за секунду, підтримкою технічних компаній та хорошою документацією. Однак відсутність практичних прикладів, книг, додаткових матеріалів та курсів, складність програм та великий об'єм коду роблять цей фреймворк важким у розумінні і роботі для початківців.

Фреймворк для розробки програмних застосунків з використанням машинного навчання. Створений в національному університеті Сінгапуру в 2014. Він зосереджений на розподіленому глибинному навчанні шляхом розбиття моделі та даних на вузли в кластері та паралелізації процесу тренування.

Проект підтримується компанією Apache, є відкритим програмним забезпеченням і розповсюджується під ліцензією Apache 2.0.

Ця бібліотека є дуже простою і на ній можна швидко реалізувати деякі елементарні мережі. Є велика кількість прикладів реалізації нейронних мереж на мові C++. Плюсом також є те, що вона включає підтримку згорткових та рекурентних мереж, забезпечуючи при цьому високу продуктивність.

Архітектура (рисунок 4.2) [26] включає три основні компоненти – ядро, модель та ввід/вивід. Ядро забезпечує керування пам'яттю та операції тензору, ввід/вивід має класи для зчитування та запису інформації з диску та мережі.

Модель надає інформаційні структури та алгоритми для машинного навчання моделей, а саме моделей нейронних мереж, оптимізаторів, метрик та втрат для основних моделей машинного навчання.

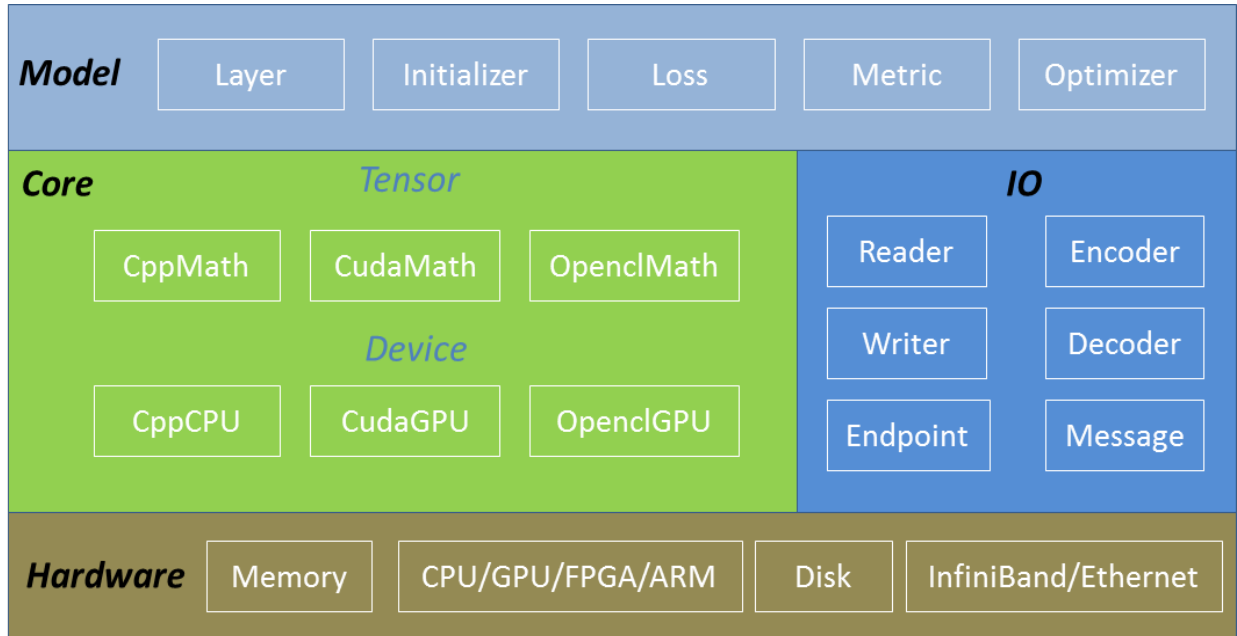


Рисунок 4.2 – Архітектура Apache SINGA

Недоліками даного фреймворку є невелика кількість прикладів на мові програмування Python, невелика спільнота, складність в установці та відсутність додаткових матеріалів, окрім документації, яка в основному представлена прикладами для мови C++.

Враховуючи велику кількість недоліків, а також відсутність повноцінної підтримки мови Python було вирішено відмовитись від використання фреймворку.

Microsoft Cognitive Toolkit, раніше відомий як CNTK, представляє собою фреймворк глибинного навчання розроблений компанією Microsoft. Він описує нейронні мережі як серію обчислювальних кроків за допомогою спрямованого графа.

Як і вищеописані фреймворки, CNTK є відкритим програмним забезпеченням для розподіленого глибинного навчання. Cognitive Toolkit дає змогу працювати у сферах розпізнавання об'єктів, голосу та пошуку релевантних результатів. Ядро CNTK реалізоване на C++ з можливістю керування на C++, C# та Python. Також

планується впровадити підтримку мови R. Тривалий час була відсутня підтримка мови Python, що було суттєвим недоліком для розробників за межами компанії Microsoft. Тому у фреймворку не сформувалось настільки сильної спільноти, як наприклад у Caffe2. З цього ж виникає і обмеження в кількості матеріалів для початківців на мові Python. Окрім документації і декількох статей важко знайти інформацію щодо програмного забезпечення.

Сильною стороною фреймворку є можливість працювати на базі звичайного ПК з використанням як CPU, так і ігрових відео-карт виробництва NVIDIA.

CNTK дозволяє легко реалізовувати та комбінувати популярні моделі типів, такі як глибинна нейронна мережа прямого поширення (feed-forward DNN), згорткові нейронні мережі (convolutional neural networks, CNN) та рекурентні нейронні мережі (RNN). Фреймворк реалізовує навчання з використанням алгоритму стохастичного градієнтного спуску з автоматичною диференціацією та паралелізацією на декілька графічних процесорів та серверів.

У березні 2018 компанія Microsoft презентувала новий фреймворк – WinML. Це набір інструментів, який дозволить розробникам використовувати всі можливості будь якого пристрою Windows 10 для підрахунків попередньо навчених моделей машинного навчання і завантажених в додатки в форматі Open Neural Network Exchange (ONNX). Архітектура застосунку приведена нижче. (рисунок 4.3). [27]

Цей продукт повинен дозволити компанії залучити нових розробників, які раніше не використовували програмне забезпечення Microsoft через його тісну пов'язаність з операційною системою Windows. Цей проект отримав підтримку від компаній технологічних гігантів, а саме – Amazon, Facebook та Microsoft.

Згідно оголошень, розробники зможуть конвертувати свої моделі в формат ONNX, після чого використовувати її в своїх додатках. До того ж розробнику не потрібно турбуватись про обладнання, що доступне на пристрої. Механізм WinML буде динамічно використовувати апаратне забезпечення і створить необхідний нативний код, щоб забезпечити максимальну продуктивність.

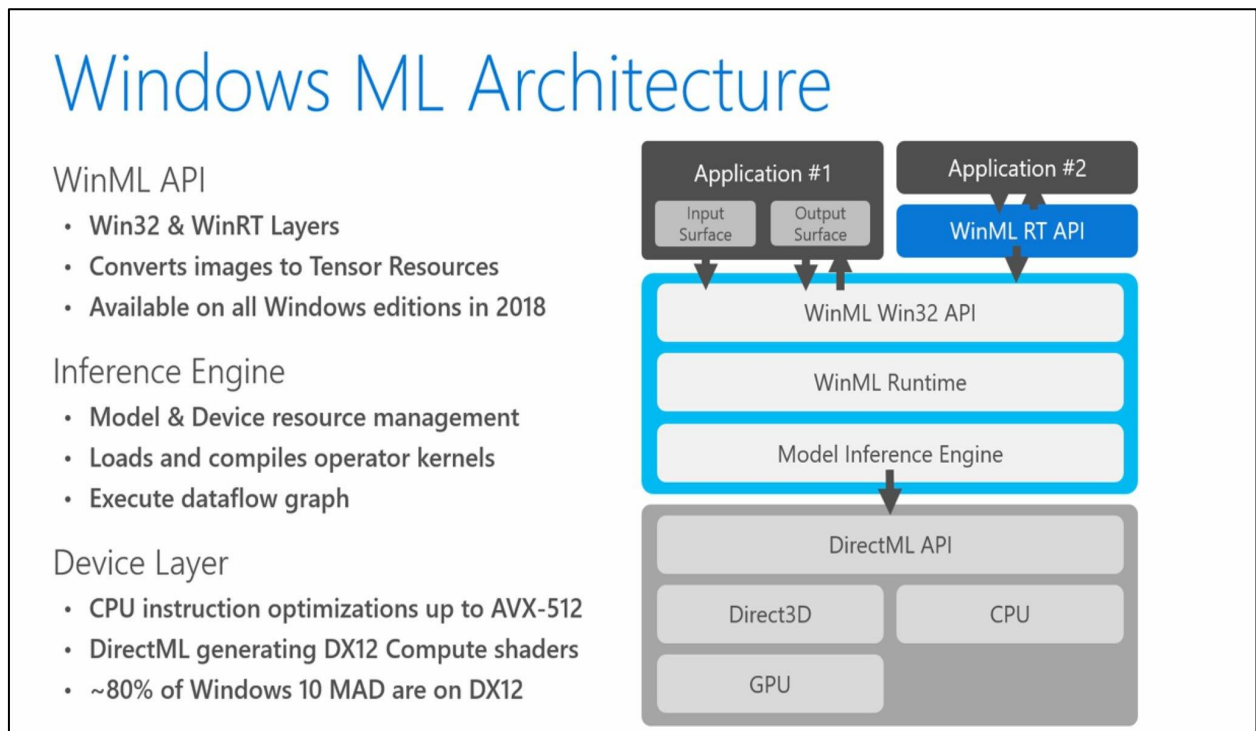


Рисунок 4.3 – Архітектура Windows ML

В результаті розробники отримують ряд переваг – низька латентність, зменшення експлуатаційних витрат та гнучкість. Однак, так як технологія зовсім нова, у неї не встигла сформуватись достатня кількість послідовників. Також від використання в реальних проектах зупиняє і мала кількість прикладів використання технології.

Загальний висновок про CNTK – фреймворк є чудовим рішенням для повсякденних задач машинного навчання. Однак це справедливо для мови програмування C++, підтримка мови Python з'явилась відносно нещодавно і фреймворк ще не встиг набрати необхідної кількості прикладів використання і практик, що і стало основною причиною відмови від вибору даного рішення.

Apache MXNet – це гнучкий та продуктивний фреймворк для глибинного навчання. Він як і решта раніше розглянутих застосунків є відкритим програмним забезпеченням і розповсюджується під ліцензією Apache2.0.

Перевагою цього фреймворку є те, що він отримав широке розповсюдження серед мов програмування, таких як C++, Python, Julia, Scala, R та Matlab. У нього

також є інструмент, що дає змогу перетворювати та перевикористовувати моделі Caffe.

Основну частину програмного забезпечення складають високорівневі прикладні інтерфейси (API). Окрім розпізнавання символів, в MXNet також передбачені всі необхідні компоненти для класифікації образів, від завантаження даних до побудови моделей з методами для початку навчання. Основні функції фреймворку поділяються на системні та користувацькі модулі (рисунок 4.4). [28]

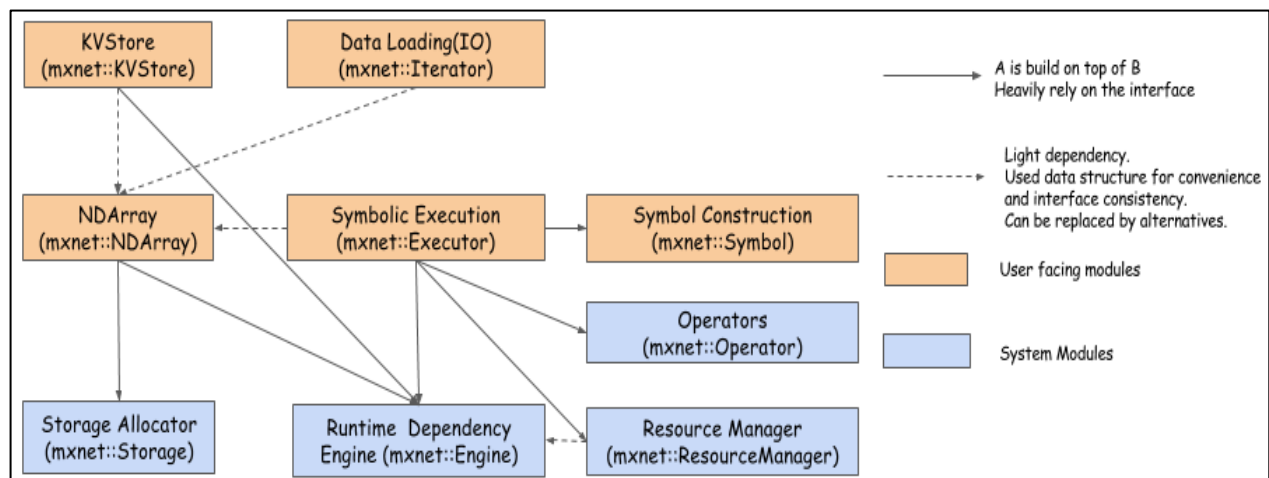


Рисунок 4.4 – Архітектура фреймворку MXNet

Плюсами фреймворку є також те, що він неперевершено справляється з розподілом роботи процесору та використання пам'яті, оскільки ці ресурси є обмеженими. Він підтримує розподілену систему обчислень, що дає змогу продуктивно тренувати нейронні мережі.

MXNet підтримує два стилі програмування – імперативний та символічний. Імперативний – це класичний стиль до розв'язування задач покроково, в той час коли символічний дозволяє визначити функції абстрактно за допомогою графіків обчислень.

Основним недоліком є підтримка дуже малої кількості низькорівневих операторів, які можна використовувати в якості складових компонентів при створенні нових моделей, не витрачаючи часу та зусиль на написання їх власноруч. Це робить складним не тільки модифікацію уже існуючих рішень, а й налаштування

під предметно-орієнтовані користувацькі дані. Також фреймворк не підтримує керування потоками, що може погіршувати продуктивність навчання в деяких випадках.

Причиною відмови від використання цього інструменту також є те, що він не призначений для розпізнавання звукових записів, а в основному зосереджений на розпізнаванні символічних даних.

Перш ніж розглянути фреймворк PyTorch варто звернути увагу на його попередника – Torch. Це відкрита бібліотека для машинного навчання, мова сценаріїв та система для наукових розрахунків на основі мови Lua. Надає великий вибір алгоритмів для глибинного машинного навчання. Він використовується в таких компаніях як Facebook, IBM та Яндекс. В останніх версіях було додано підтримку для використання під мобільну розробку. [29]

Основними функціями є потужний багатовимірний масив, багато методів індексування, зрізів та транспонування даних, алгоритми лінійної алгебри, підтримка нейронних мереж, оптимізатори для числових розрахунків, швидке та ефективно використання відеокарти.

Основною ціллю Torch є надання користувачеві максимуму гнучкості та швидкості в розробці наукових алгоритмів, в той же час цей процес є надзвичайно простим. Фреймворк подається з потужною екосистемою підтримуваною спільнотою (рисунок 4.5). [29] В його арсеналі можна знайти багато застосунків, які допомагають вирішити основні задачі машинного навчання. Забезпечується підтримка наступних пакетів: машинного навчання, комп'ютерного бачення, обробки сигналів, паралельних обчислень, обробка аудіо- та відеофайлів, а також зображень. Всі вони підтримувались та розроблялись спільнотою мови Lua.

PyTorch – відкрита бібліотека для мови програмування Python для машинного навчання, базована на Torch, використовується для таких додатків, як обробка природньої мови (Natural Language Processing, NLP). Розроблялась у науковій групі ШІ компанії Facebook. На основі цього фреймворку розроблено програмне забезпечення для ймовірнісного програмування «Руго» компанії Uber. [29]

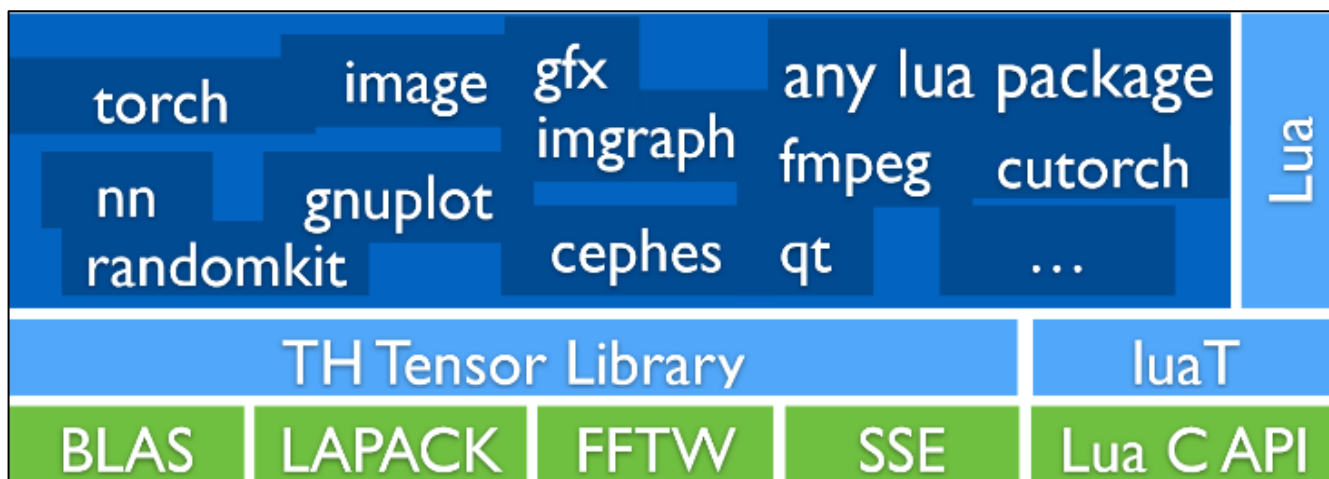


Рисунок 4.5 – Архітектура фреймворку Torch

Основними відмінностями Torch та PyTorch є те, що Torch – це тензорна бібліотека як NumPy, однак на відміну від нього він має сильну підтримку GPU. Lua виступає обгорткою для Torch, тому необхідно мати знання цієї мови. Також для роботи знадобиться пакетний менеджер LuaRocks. Для використання PyTorch немає потреби у вивченні мови Lua та установці додаткових пакетних менеджерів. Можна використовувати усі переваги мови Python, а саме його розвинене середовище та бібліотеки на зразок SciPy, NumPy та matplotlib.

Перевагами PyTorch є те, що у ньому доволі легко розширювати класи. Забезпечено прості приклади по архітектурі та налаштуванню, хоча вони і об'ємні. У PyTorch сильна спільнота, яка забезпечує швидке вирішення проблем.

Недоліками ж є те, що документація відстає від функціональності, а це означає те, що іноді доводиться переглядати вихідний код бібліотеки. Немає подібного інструменту до TensorBoard, немає вбудованих обгортки для функцій зворотнього виклику (callback functions). Якщо і є якісь напрацювання то вони ще не готові для повноцінного використання і знаходяться в стадії активної розробки.

Висновок – фреймворк має сильні сторони, однак ще не повністю готова для використання в реальних продуктах, до того ж, для того щоб її використовувати необхідно мати сильні знання в області машинного навчання, тому що є ймовірність того, що щось доведеться дописувати вручну, а це означає зайві витрати часу на розробку того, що уже представлено у інших фреймворках одразу.

9 листопада 2015 року компанія Google презентувала бібліотеку для машинного навчання TensorFlow. Вона була розроблена з метою забезпечити потреби розробки систем, що здатні будувати та тренувати нейронні мережі для передбачення, виявлення образів, розпізнавання об'єктів, обробки та аналізу великої кількості даних. Процеси тренування та розпізнавання дужі схожі до тих які використовує людський мозок.

Як і попередні фреймворки TF є відкритим програмним забезпеченням, випущеним під ліцензією Apache2.0. Проект реалізовано в основному мовами Python та C++. Він підтримується усіма поширеними платформами – Linux, macOS та Windows. Використання TF підтримується багатьма мовами, а саме Python, C++, Java, Go, Swift та JavaScript. Також спільнота розробила API, що дозволяє використовувати TF з багатьма іншими мовами, наприклад C#, Haskell, Rust та Rubi.

Наразі TF є лідером галузі машинного навчання, це підтверджується і пошуковими запитами користувачів (нисунок 4.6). [30] З моменту виходу першої версії у нього сформувалась хороша спільнота, було написано багато статей, розроблено багато допоміжних інструментів, що стало великою перевагою.

Варто зазначити декілька фактів про Keras. Це нейромережева бібліотека, що здатна працювати поверх TF та Theano. Тобто використовуючи TF можна використовувати всі переваги бібліотеки Keras без будь яких додаткових установок чи налаштувань.

Ще один популярний фреймворк який не розглядався більш детально – Theano. З релізом TF популярність Theano поступово зменшувалась, оскільки забезпечують підтримку бібліотеки Keras, і Theano забезпечувала розробку лише за допомогою мови Python, і з вересня 2017 року команда Theano оголосила про зупинку розробки фреймворку. Нараз цей фреймворк використовують лише для підтримки уже існуючих продуктів. [25]

TF дуже простий в інсталяції. Його архітектура складається з моделей (models), оцінювачів (estimators) та шарів (layers). В свою чергу шари будують моделі, а оцінювач навчає та досліджує точність моделі.

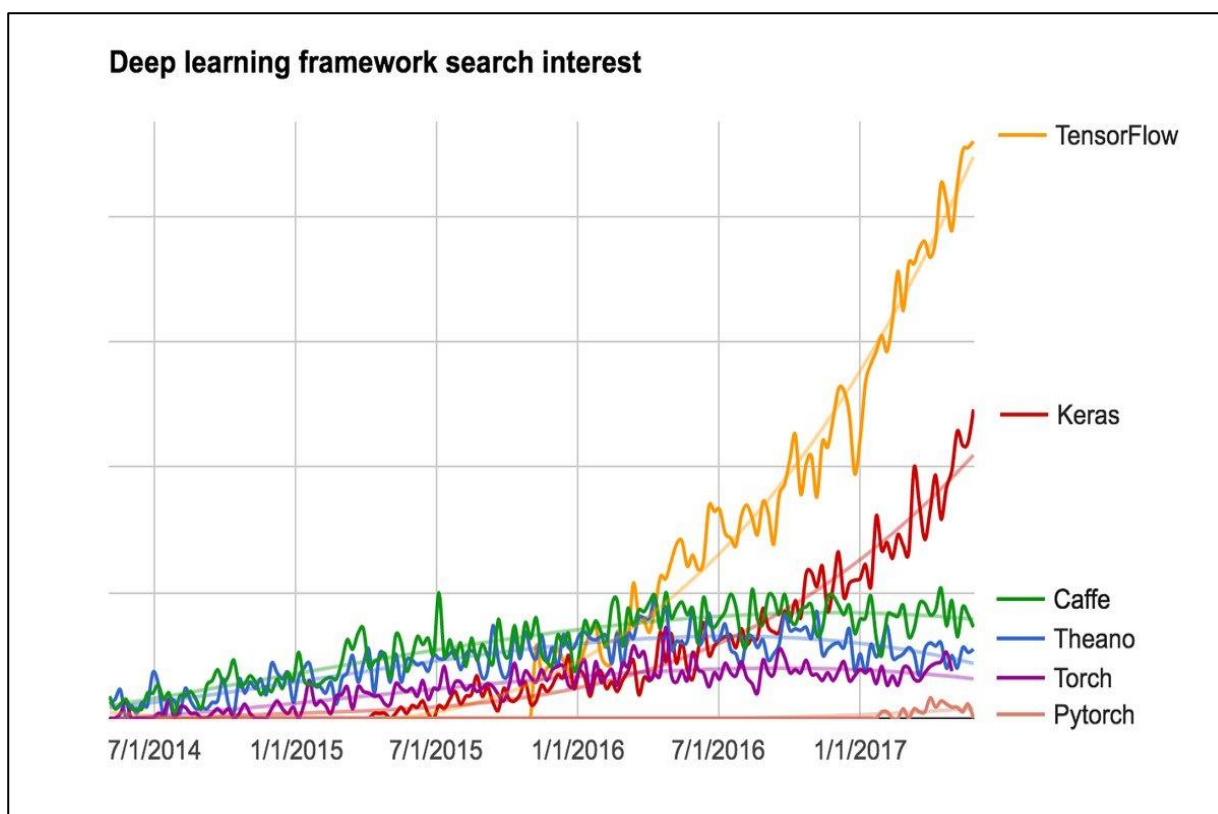


Рисунок 4.6 – Порівняння пошукових запитів фреймворків для машинного навчання

Також можна підключати бази даних для маніпуляцій з даними (додавання, видалення, редагування та збереження результатів). На більш низькому рівні знаходиться двигун, який відповідає за розподілене виконання процесів фреймворку (рисунок 4.7). [31] Варто зауважити, що користувачеві не доводиться працювати з рівнем двигуну незалежно від обраної мови. На нижніх шарах знаходяться CPU та GPU. TF порівняно забезпечує середній рівень продуктивності (приблизно у 2 рази гірше ніж MXNet).

Перевагою TensorFlow є також підтримка мобільних пристроїв на базі iOS та Android, що допомагає залучити розробників з цієї галузі. Маючи великий набір низькорівневих операторів він дає змогу налаштовувати та тренувати модель так як цього потребує користувач.

Одним з недоліків є неможливість перенесення моделей розроблених на інших фреймворках, що робить можливим лише створення нейронної мережі з нуля. Однак він не має ніякого значення для нових проектів.

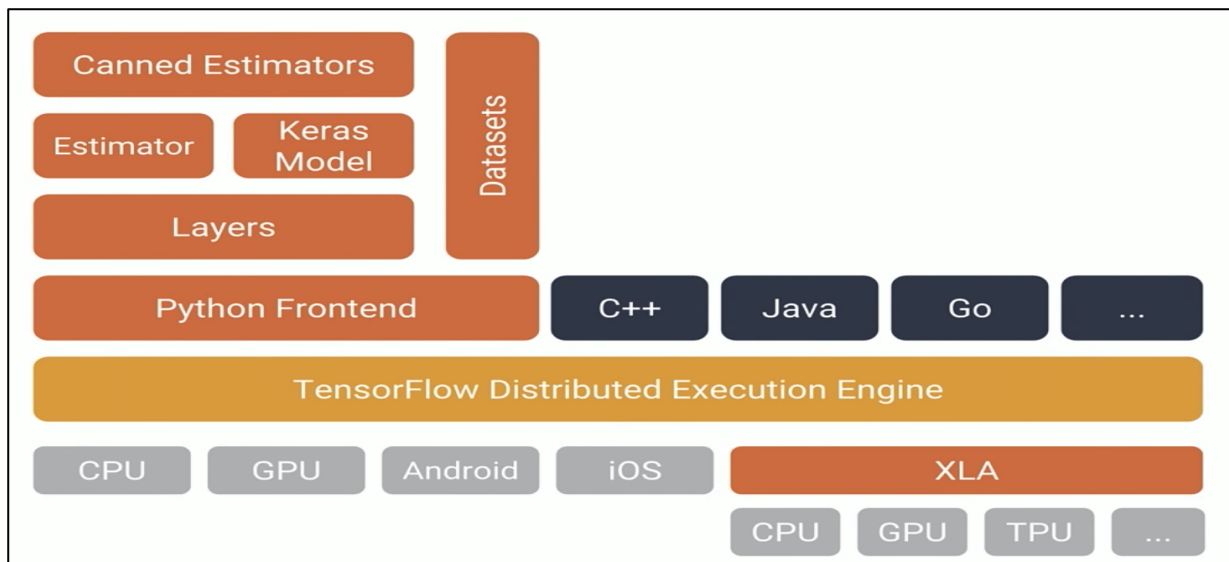


Рисунок 4.7 – Архітектура TensorFlow

Для того щоб оцінювати і відслідковувати навчання нейронної мережі варто встановити додатковий інструмент TensorBoard. Після цього потрібно вказати відповідну модель і запустити сервіс, що покаже характеристики нейронної мережі, наприклад, точність, ентропію, похибку, граф тренувальної моделі та багато іншого. Інструмент дає змогу користувачеві налаштувати необхідні параметри у простому та зручному інтерфейсі (рисунок 4.8). [31]

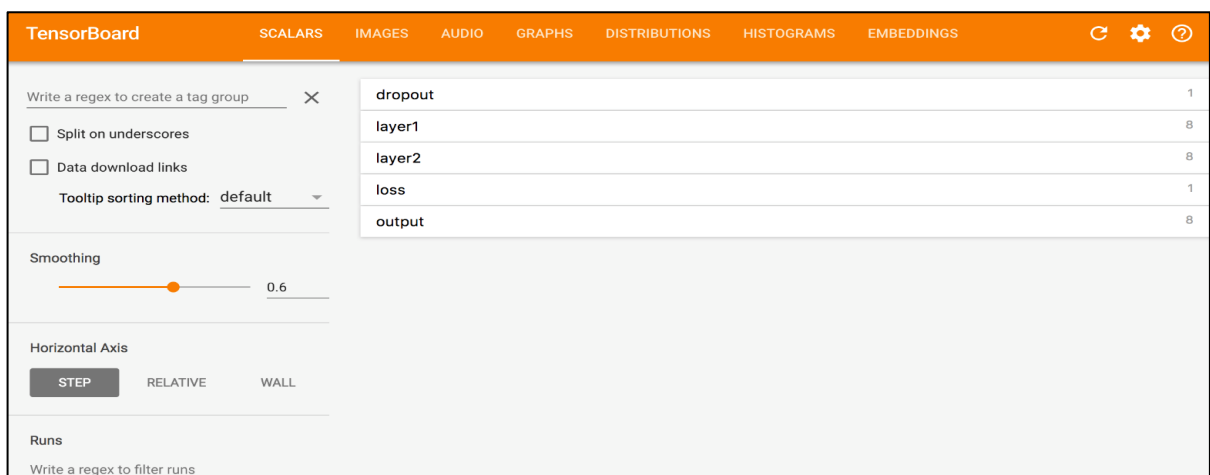


Рисунок 4.8 – Інтерфейс TensorBoard

Враховуючи усі переваги даного фреймворку, а саме можливість легко розпочати працювати з фреймворком, доступність різноманітних інструментів, що

полегшують розробку, зручне сучасне API, наявність хорошої спільноти та різноманітних додаткових матеріалів вплинуло на остаточний вибір цього TF як основного фреймворку для розробки програмного продукту.

Висновки до розділу 4

У даному розділі було розглянуто важливість вибору правильного фреймворку, було розглянуто важливі аспекти, які слід враховувати під час вибору, а також було проведено аналіз сучасних популярних програмних засобів.

Представлено огляд базових репозиторіїв зразків музичних інструментів. Виокремлено значимі характеристики для їх порівняння. На основі проведеного огляду для подальшого формування навчальної вибірки класифікатору музичних інструментів було обґрунтовано вибір бібліотеки Лондонського симфонічного оркестру (London Philharmonic Orchestra Dataset).

Розглянуто 3 найпоширеніші бібліотеки для обробки аудіосигналів – aubio, Yaafe та LibROSA. Проведено аналіз їхніх функцій та зручності у використанні, у результаті якого вибір було зроблено на користь бібліотеки LibROSA через її простоту у використанні та велику кількість прикладів використання.

До найважливіших факторів, які впливають на вибір інструментів відносяться доступність навчальних матеріалів, популярність, основні особливості та можливості інструменту, простота у використанні та інтеграції з іншими сервісами.

До найпопулярніших фреймворків відносяться PyTorch, MXNet, Apache SINGA, Caffe2, Microsoft Cognitive Toolkit та TensorFlow.

MXNet, Apache Singa та PyTorch є хорошими засобами, але поки що не отримали великої підтримки у спільноти через свою нещодавню появу.

Microsoft Cognitive Toolkit забезпечує хороші можливості, має велику спільноту, однак розрахований у більшій мірі на розробників C++, і лише з недавнього часу почав підтримувати мову Python.

Caffe2 дозволяє працювати з зображеннями та символами, забезпечуючи при цьому високу продуктивність, має хорошу підтримку з боку технічних компаній.

Проте, для початківців він є досить складним через невелику кількість додаткових матеріалів, складність програм та великий об'єм.

TensorFlow є беззаперечним лідером ринку в галузі машинного навчання. Він надає зручне API, підтримується компанією Google та широкою аудиторією програмістів. Для цього фреймворку досить легко знайти статті, відеоматеріали чи курси. Він забезпечує підтримку зручних інструментів та бібліотек на зразок Keras чи TensorBoard. Його основні недоліки – середня продуктивність та неможливість повторного використання моделей розроблених на інших фреймворках, однак вони несуттєві враховуючи переваги.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ

У цьому розділі розглядається використання обраних інструментів для вирішення задачі класифікації музичних інструментів.

Задача полягала у вирішенні наступних підзадач:

- екстракції характеристик для прикладів гри музичних інструментів;
- побудові та тренуванні нейронної мережі MLP;
- побудові та тренуванні нейронної мережі CNN;
- візуалізації отриманих результатів та проведенні порівняльної характеристики двох алгоритмів.

У розділі розглядаються такі питання, як побудова архітектури нейронної мережі, виконання сценаріїв роботи окремих модулів та основні інструменти, що допомагали у вирішенні конкретних завдань.

5.1 Архітектура програмного додатку

Відповідно до обраних програмних засобів архітектура програмного рішення представлена на рисунку нижче (рисунок 5.1):

На першому етапі, з бази даних дістається певний аудіо-запис. Далі для нього відбувається екстракція характеристик, які передаються як вхідні дані до нейронної мережі прямого поширення. В загальному відбувається екстракції характеристик усіх музичних інструментів і отримані результати зберігаються до csv файлу. Ці дані потім використовуються для тренування, тестування та візуалізації навчальної вибірки.

Потім відбувається процес розробки архітектури нейронної мережі для того щоб побудувати модель розпізнавання звучання музичних даних. Він тренується та тестується на відповідних підготовлених вибірках. В результаті класифікації звучання виводиться результат аналізу, а саме конкретний інструмент, що було визначено для поданого музичного запису. Записи на вхід подаються у форматі mp3, після виконується екстракція характеристик.

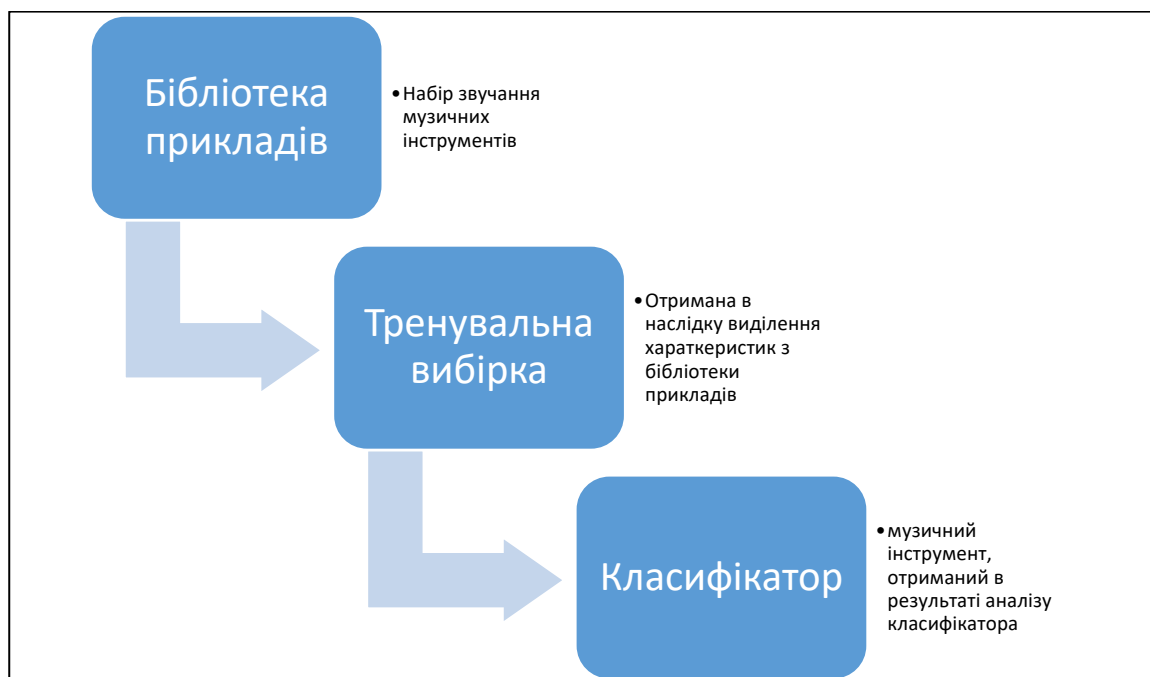


Рисунок 5.1 — загальна архітектура системи

Загальний підхід до розв’язання задачі полягає у вирішенні певних етапів. Спочатку необхідно знайти бібліотеки прикладів та обрати ту, яка підходить найкраще для даного випадку. В роботі було обрано бібліотеку записів лондонського симфонічного оркестру.

Для виокремлення музичних характеристик було обрано бібліотеку LibROSA.

Для розробки нейронних мереж було обрано фреймворк TensorFlow.

Важливою складовою програмного продукту є візуалізація даних, яка забезпечується за допомогою бібліотек TensorBoard та Matplotlib.

5.2 Формування навчальної та тренувальної вибірки

Екстракція характеристик здійснюється за допомогою бібліотеки LibROSA, її опис приводився раніше (підрозділ 4.4). Для того щоб отримати дані для тренувальної та тестової вибірки потрібно завантажити екземпляри гри музичних інструментів у форматі mp3 до папки /samples. Після цього командою виконати сценарій екстракції характеристик – feature_extractor.py. Після виконання сценарію користувач отримає повідомлення про успішне виконання операції, а дані зберігаються у форматі CSV.

У результаті екстракції було отримано 7294 записи, які містять інформацію про 8 різних інструментів. 879 – флейта, 890 – віолончель, 973 – туба, 1503 – скрипка, 653 – валторна, 832 – тромбон, 853 – контрабас, 721 – фагот. 80% отриманих даних було виділено на етап тренування, а 20% – тестову вибірку.

Для візуалізації отриманих даних (рисунок 5.2, рисунок 5.3) використовується бібліотека Matplotlib та алгоритми зменшення розмірності PCA та t-SNE. Вибірку для візуалізації створено на основі випадкових 100 екземплярів звучання кожного інструменту.

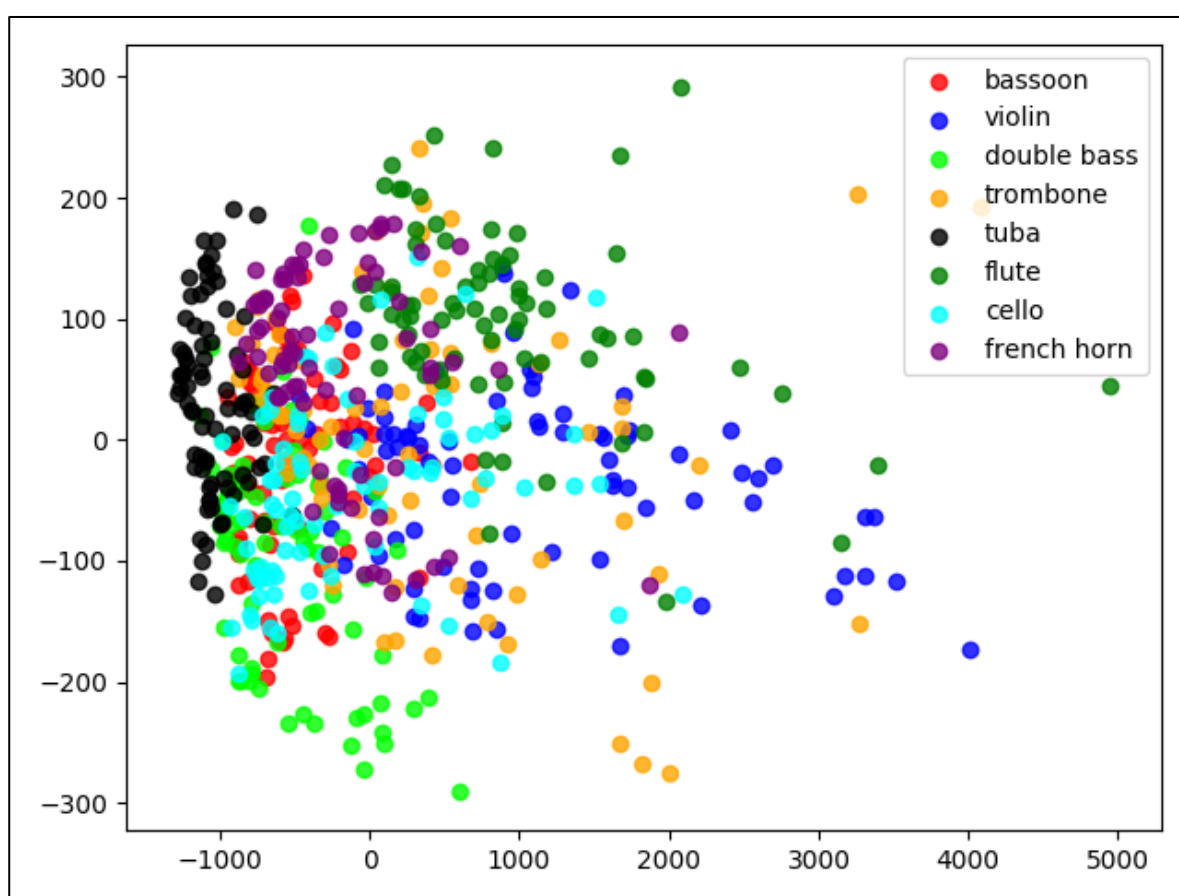


Рисунок 5.2 — отримані дані за допомогою PCA

Для візуалізації графіку PCA було відібрано по 100 записів гри кожного інструменту, оскільки під час візуалізації усієї навчальної вибірки малюнок важко аналізувати, тому що деякі інструменти повністю перекривають дані інших інструментів.

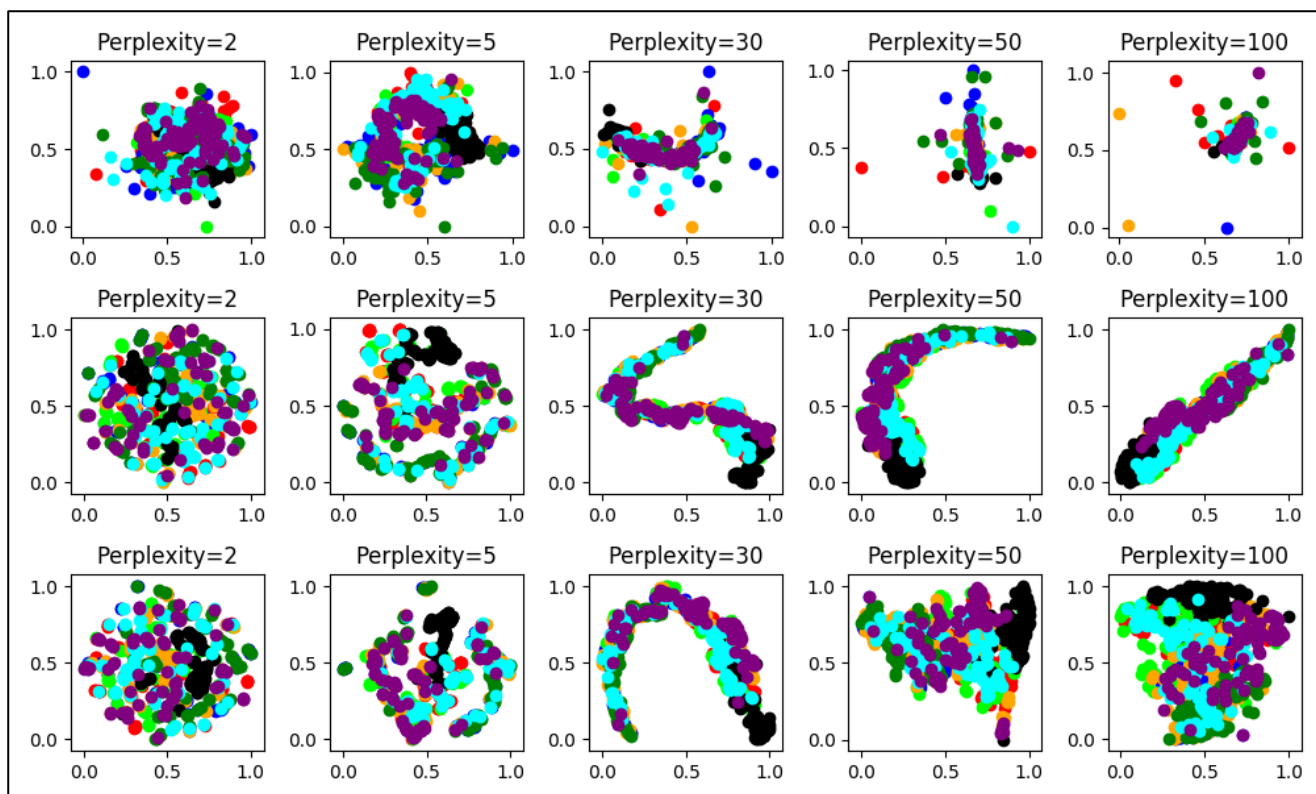


Рисунок 5.3 — отримані дані в результаті перетворення t-SNE

Як видно з графіків, дані частково відокремлюються і групуються у певні сімейства, однак досить складно зробити остаточний висновок з положення точки у просторі. Отже, задачу важко розв'язати за допомогою звичайних лінійних алгоритмів, а тому доцільним є використання методів штучного інтелекту, а саме нейронних мереж.

5.3 Вирішення задачі за допомогою MLP

Мережа складається з 2 шарів по 16 нейронів кожен. Функція активації в кожному нейроні – ReLU. Оптимізатор – RMSProp. Навчання відбувається з допомогою класичного методу зворотного розповсюдження і стохастичного градієнтного спуску.

Вхідними даними системи є обрані в першому розділі характеристики:

- $\{ X_1, \dots, X_{20} \} = \text{MFCC features}$
- $\{ X_{21} \} = \text{Spectral centroid}$

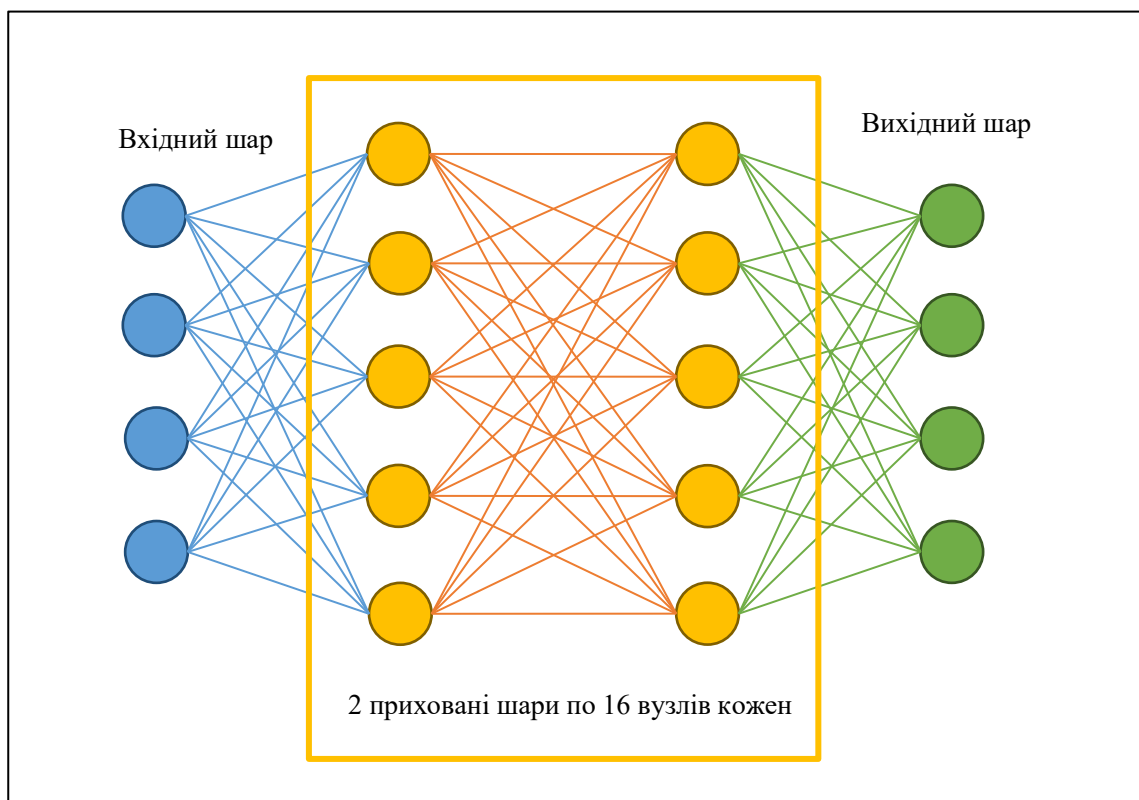


Рисунок 5.4 — Архітектура багатошарової нейронної мережі

На виході – 8 нейронів, кожен з яких відповідає за певний інструмент.

Цей класифікатор створюється та навчається в модулі `neural_network.py`. В кінці навчання класифікатор серіалізується за допомогою бібліотеки `pickle`, для того щоб навчена нейронна мережа не перенавчалась для операції розпізнавання музичного інструменту, і зберігається до файлу `model.pickle`.

Класифікатор нейронної мережі навчається на основі 1964 прикладів гри 8 музичних інструментів. В результаті цього процесу створюється навчений класифікатор, який здатний розпізнавати музичні інструменти. Точність визначення складає 90.2%. Після того як класифікатор навчений виконується передбачення для тестової вибірки з 1435 прикладів.

Візуалізація процесу навчання відбувається за допомогою розширення `TensorBoard`. Цей засіб надає змогу переглядати графіки функції втрат (рисунок 5.5) та функції точності (рисунок 5.6).

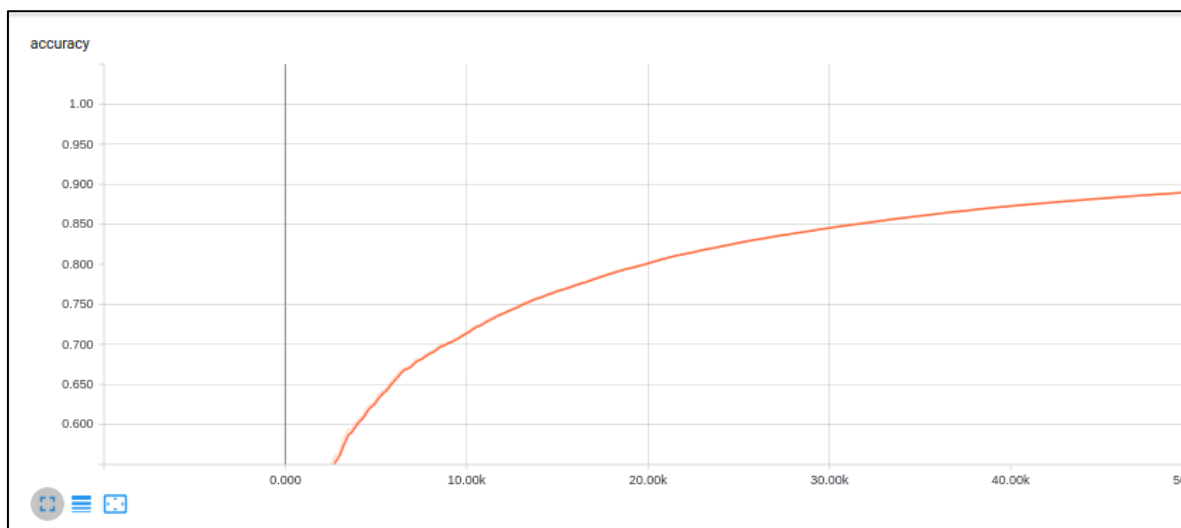


Рисунок 5.5 — графік навчання MLP

Аналізуючи отримані графіки можна визначити оптимальні параметри кількості кроків, для того щоб уникнути процесу перенавчання, коли з подальшим навчанням результат точності тільки погіршується. Також можна аналізувати процес навчання, який без подібних графіків стає подібним до роботи чорного ящика.

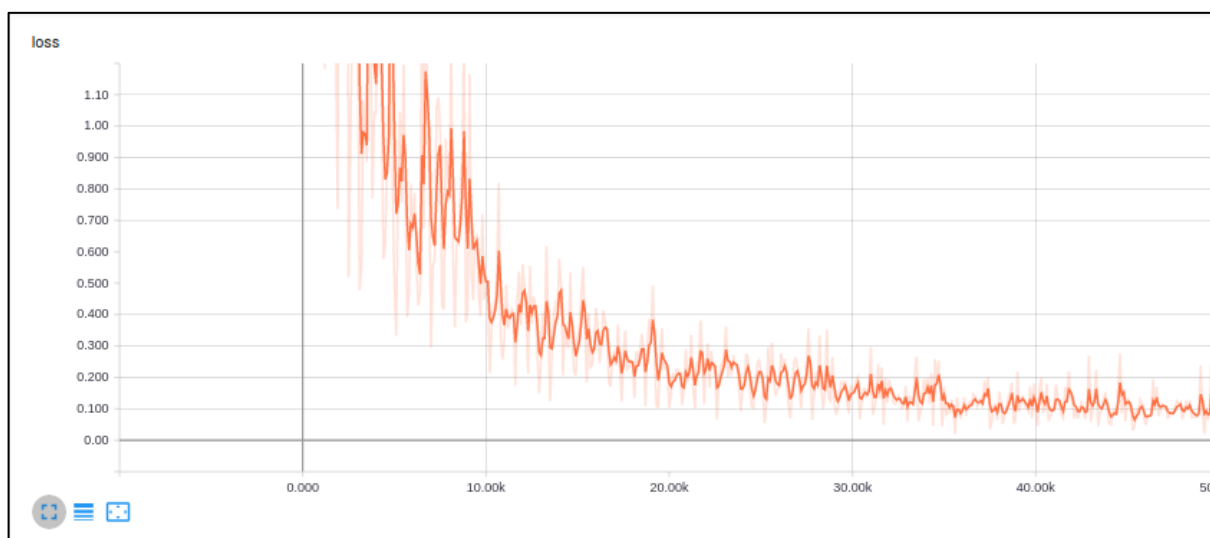


Рисунок 5.6 — графік похибки MLP

Як видно з графіку функції втрат в залежності від кроку — з подальшим навчанням значення зменшується, а значення графіку точності обернено пропорційно зростає досягаючи результату в 90.2%.

5.4 Вирішення задачі за допомогою CNN

В роботі пропонується використати в якості вхідного шару для мережі CNN отримані характеристики для музичних записів довжиною в одну секунду, замість того щоб використовувати спектрограму. Тобто вхідним шаром класифікатору є дані розмірністю 20, 42, 1, де 20 – кількість мел-кепстральних характеристик, 42 – тривалість музичного запису, що складає 1 секунду. Остання розмірність рівна 1, оскільки в якості характеристик використовується лише одна характеристика на момент часу, на відміну від зображень, у яких ця величина рівна 3, оскільки кожен піксель складається з трьох основних кольорів – червоного, синього та зеленого.

Загальна архітектура представлена на рисунку 5.7.

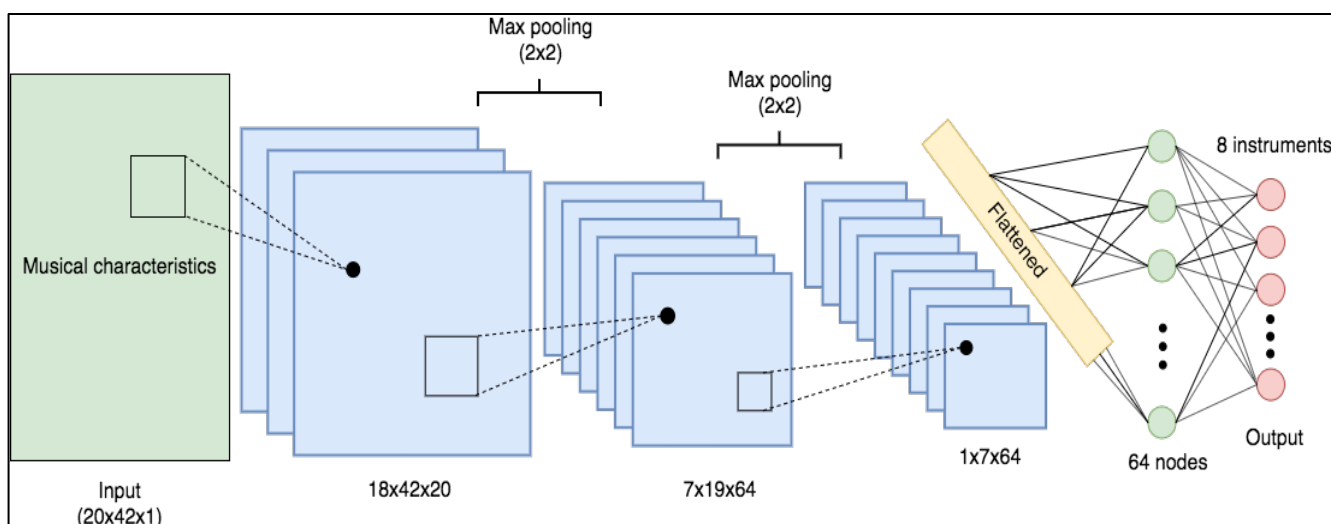


Рисунок 5.7 — архітектура CNN класифікатору

З рисунку зрозуміло, що згорткова нейронна мережа містить 3 згорткових шари різної розмірності, 2 шари максимального об'єднання. В кінці згорткової мережі виконується етап приведення отриманих даних до двовірного простору. Ці дані слугують вхідним шаром для глибокої нейронної мережі, яка містить 1 прихований шар, в якому знаходиться 64 вузли. На виході, як і у MLP CNN має вузол, що відповідає 8 музичним інструментам обраних для експерименту.

В якості функції активації між шарами використовується функція ReLU, окрім останнього шару – там використовується функція SoftMax.

5.5 Отримані результати

Розробивши нейронні мережі на основі багатошарової та згорткової нейронних мереж можна починати процес навчання.

Графіки навчання (рисунок 5.8) дозволяють зрозуміти наскільки якісно проходить процес навчання, дозволяє виявити явища перенавчання.

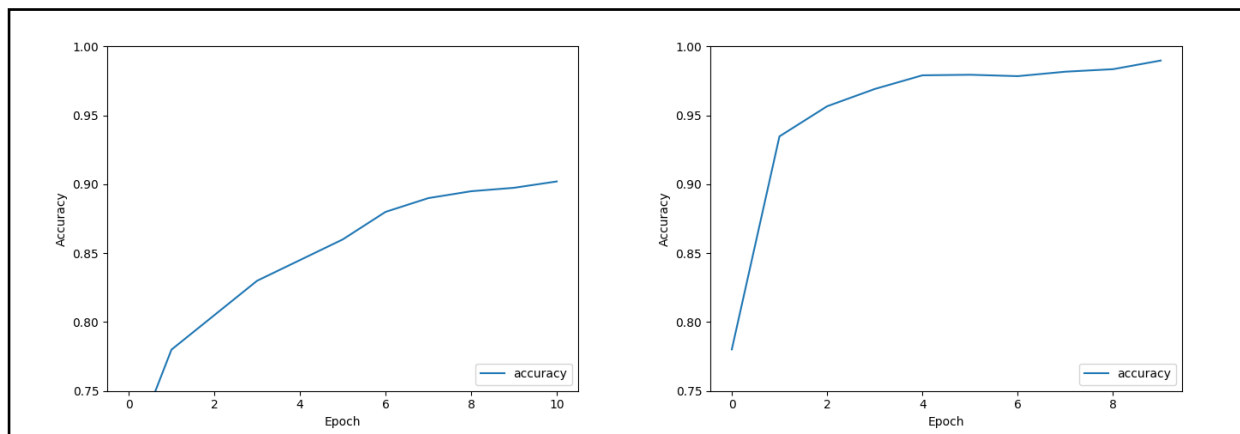


Рисунок 5.8 — графіки процесу навчання MLP (зліва) та CNN (справа)

Як видно з графіків, процес навчання обох класифікаторів проходить поступово, без різких стрибків між показниками точності.

Для більш детального аналізу процесу навчання нейронної мережі використовують додаткові інструменти, наприклад TensorBoard. Вона дозволяє отримати дані що подаються на вхідний шар, переглянути дані які потрапили в тренувальний набір, оскільки навчання відбувається не одразу, а поступово. Для цього дані підбираються випадковим чином з тренувальної вибірки. Від цього також може залежати якість навчання, однак якщо дані будуть підібрані якісно, це має відігравати дуже незначну роль і експерименти мають давати схожі результати. В процесі виконання дослідження було проведено багато експериментів, і результати були в діапазоні допустимої похибки в 1%. Можна зробити висновок, що вибірка підібрана якісно.

Якість класифікації звучання музичних інструментів зазвичай представлена за допомогою матриці колізій (рисунки 5.9 та 5.10).

Матриця колізій складається з рядків та стовпчиків, які відповідають за певний інструмент. Рядок представляє очікуваний результат, а стовпчик – отриманий. Хороша матриця колізій має вигляд діагональної матриці.

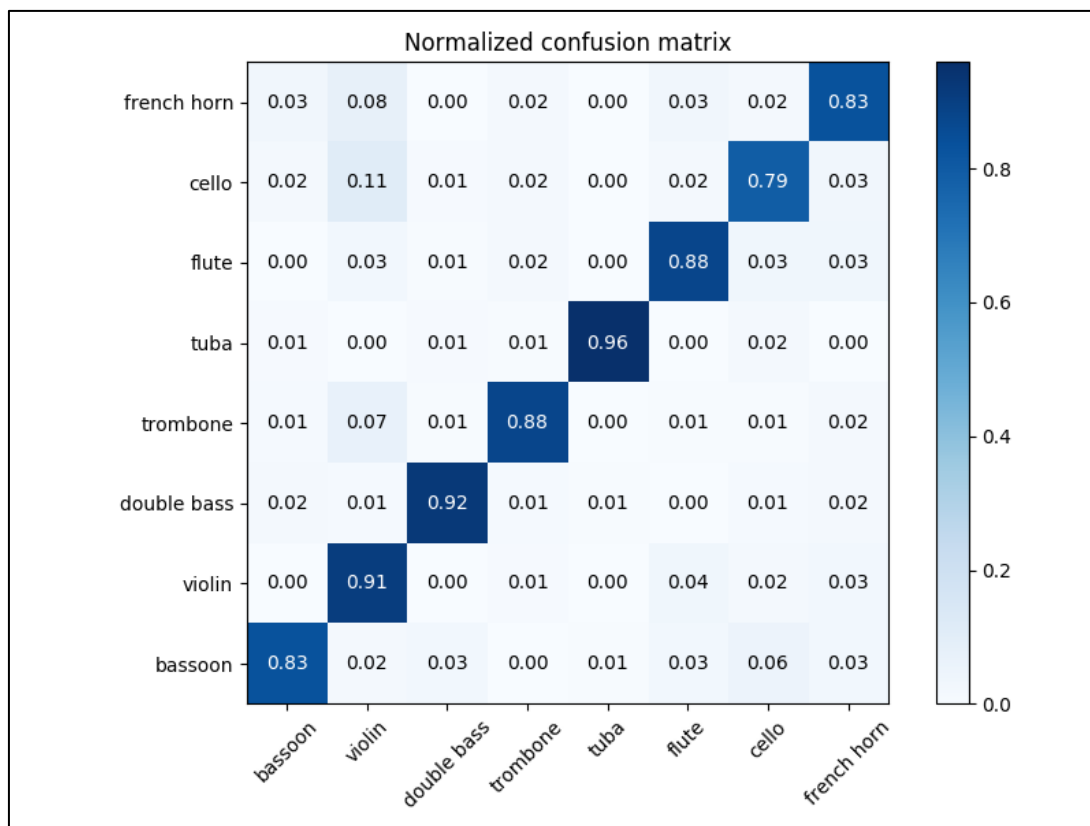


Рисунок 5.9 — матриця колізій для класифікатору MLP

Як видно з матриці колізій для MLP – класифікатор має хороший результат роботи на тестовій вибірці. Загальний результат роботи класифікатора складає 90.2%. Найкраще розпізнається туба, з результатом – 96%, а найгірший для віолончелі – 79%. Однак, відхилення в розпізнаванні віолончелі припадає на інший інструмент смичкової групи – скрипку, які є подібними у своєму звучанні, тому подібне відхилення є передбачуваним і допустимим, що також підтверджується графіком представлення навчальної вибірки (рисунок 5.2).

Порівнюючи отримані результати з попередніми дослідженнями можна прийти до висновку, що розроблений класифікатор на основі MLP показує дуже хороші результати.

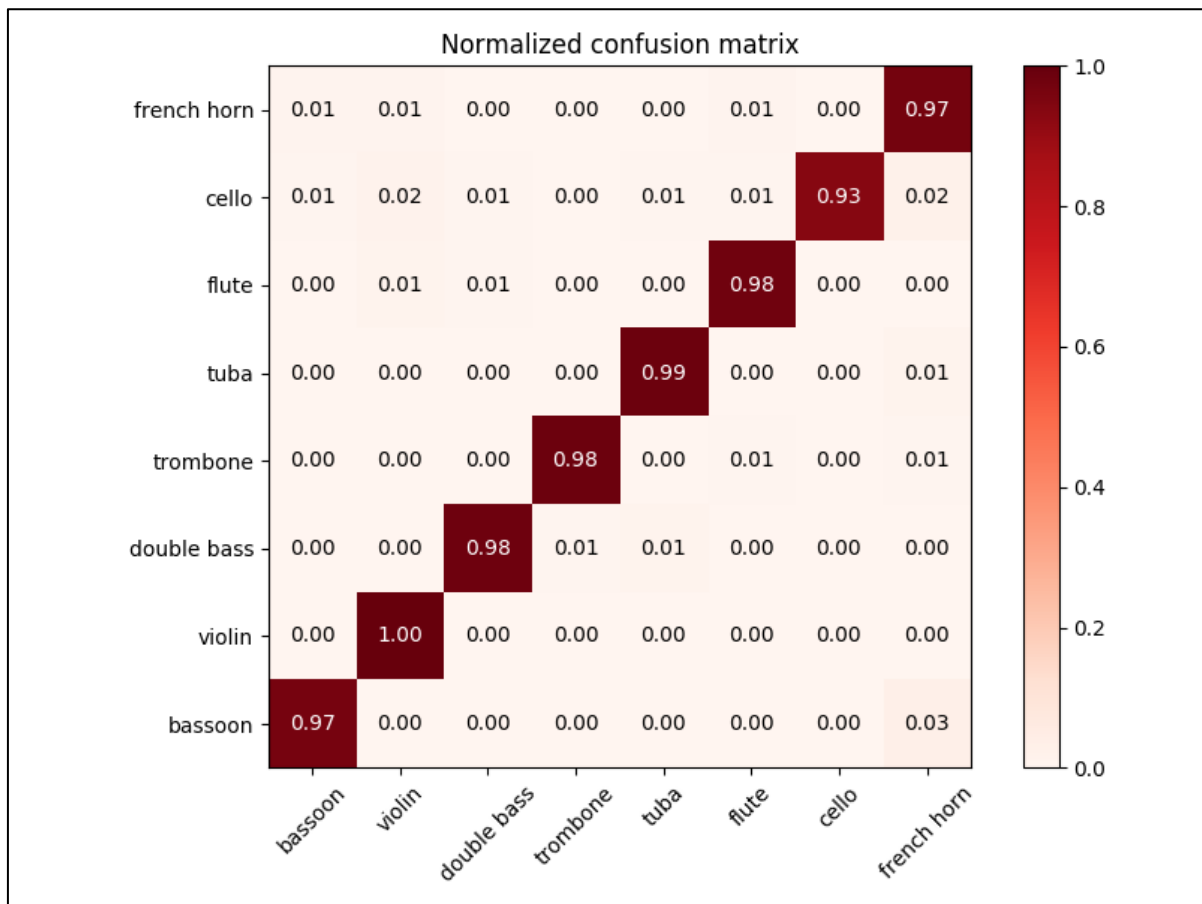


Рисунок 5.10 — матриця колізій для класифікатору CNN

Матриця колізій роботи класифікатору на основі мережі CNN також показує дуже хороші результати. Скрипку вдалось розпізнати з тестової вибірки з 100% результатом, найгірше класифікатор працює для розпізнавання віолончелі – 93%. Загальний результат роботи класифікатора складає 97.9%, що є дуже хорошим результатом у порівнянні з попередніми дослідженнями. Використання на вхідному шарі згорткової нейронної мережі музичних характеристик без приведення їх до вигляду спектрограми не погіршило результатів, тому можна прийти до висновку, що перетворення даних отриманих в результаті екстракції не потрібне. В результаті цього, мережа буде навчатись швидше, оскільки не потрібна попередня обробка зображення, як це зазвичай робиться для розпізнавання візуальної інформації.

Порівнюючи результати роботи розроблених класифікаторів можна прийти до висновку, що використання згорткових нейронних мереж краще підходить для задач розпізнавання звучання музичних інструментів ніж використання глибинних нейронних мереж.

5.6 Інтерфейс користувача

Розпізнавання музичного запису відбувається за допомогою класифікатору нейронної мережі прямого поширення отриманого експериментально. Для користувацької взаємодії з додатком було передбачено написання простого web-інтерфейсу, який реалізовано за допомогою web-фреймворку Flask. Користувач системи може завантажити бажаний файл і побачити результат класифікації на екрані.

Для того щоб визначити, який інструмент грає на аудіо-записі потрібно натиснути кнопку «Choose File» (рисунок 5.11).



Рисунок 5.11 — інтерфейс вибору файлу

Після чого з'явиться вікно вибору файлу на комп'ютері. Інтерфейс даного вікна може дещо відрізнятись в залежності від операційної системи, для Mac OS він має вигляд приведений на рисунку нижче (рисунок 5.12). Для інших систем, інтерфейс вибору музичного запису може дещо відрізнятись. Наприклад на рисунку 5.13 зображено інтерфейс для користувачі OS Ubuntu.

Потрібно обрати бажаний файл у форматі mp3 та натиснути кнопку «Open». Для прикладу пропонується вибрати файл violin.mp3, який містить приклад звучання гри на скрипці.

Додаток підтримує лише обробку аудіосигналів у форматі mp3, однак цю проблему можна легко вирішити створивши ще один інструмент для виділення характеристик, який підтримуватиме бажаний формат.

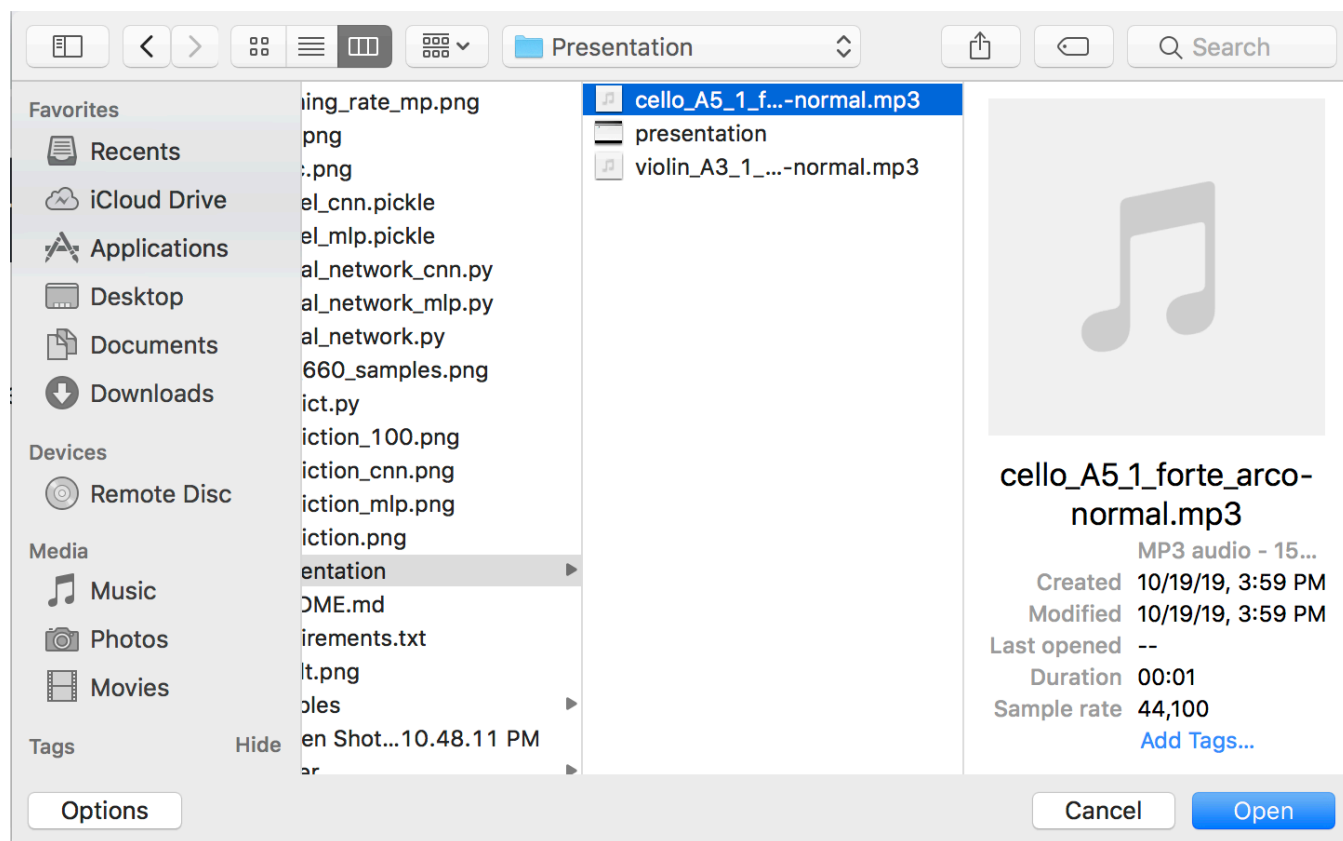


Рисунок 5.12 – Інтерфейс вибору файлу на комп'ютері

Обравши файл потрібно натиснути кнопку «Analyze», яка запустить процес аналізу звучання музичного запису з обраного файлу. На сервері зберігається копія файлу, проводиться екстракція музичних характеристик з нього і після цього інструмент який грає класифікується. Для того щоб отримати раніше навчений класифікатор музичних інструментів потрібно десереалізувати файл `model_mlp.pickle` та `model_cnn.pickle`, які містять моделі MLP та CNN класифікаторів відповідно. Далі функція прогнозування використовує метод `predict` класифікатора, що повертає результат роботи визначення музичного інструменту. Результат повертається у вигляді JSON об'єкту, в якому ключем є вид класифікатора, а значенням – результат розпізнавання. Також можна отримувати дані у вигляді списку класифікації і результату, які показали усі вихідні вузли.

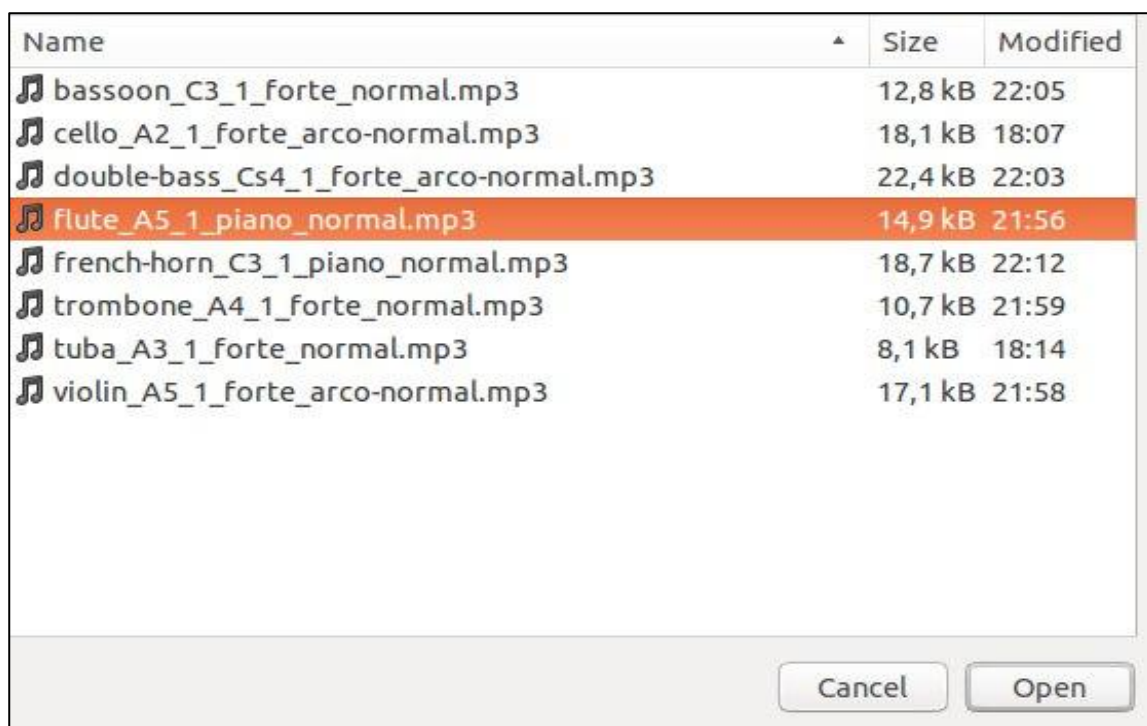


Рисунок 5.13 – Інтерфейс вибору файлу на Ubuntu OS

Недоліком системи є те, що якщо результат класифікації поганий, користувач не отримає повідомлення про помилку, а просто отримає повідомлення з % ймовірності вказаного інструменту. Отримані дані відображаються користувачеві на сторінці інтерфейсу (рисунок 5.14).

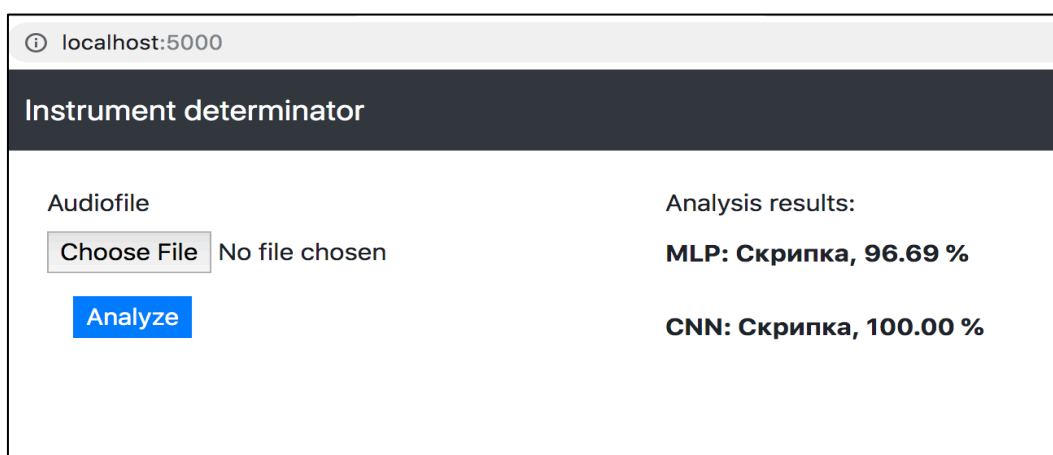


Рисунок 5.13 — приклад роботи двох класифікаторів

За даними отриманими в результаті роботи класифікаторів, інструмент, який використовується в аудіо-записі violin.mp3 – скрипка, як і очікувалось. Ймовірність

правильного передбачення для даного випадку складає 96.69% для MLP та 100.00% – CNN, що є дуже правильним результатом.

Висновки до розділу 5

У цьому розділі було розглянуто програмну реалізацію проекту та отримані результати. Розглянуто архітектуру нейронної мережі, розроблено модуль екстракції характеристик з аудіо-запису, а також класифікатори музичних інструментів, що працює з середньою точністю 90.2% – MLP та 97.9% – CNN. Отримані результати підтверджуються минулими роботами і мають кращі показники ніж були отримані в попередніх дослідженнях.

В кінці розділу продемонстровано роботу користувацького інтерфейсу для визначення музичного інструменту.

6 РОЗРОБКА СТАРТАП ПРОЕКТУ

З самого початку дана система не задумувалась як стартап проект, однак системи автоматичної ідентифікації звуків є досить популярними в наш час. Ідея полягає в розробці продукту, що забезпечуватиме ідентифікацію та аналіз гри музичних інструментів. Використовуючи сучасні технології, а саме технології штучного інтелекту, які розвиваються в останні роки, а також високу продуктивність обчислювальних машин можна створити застосунок, який буде надавати високу точність результатів класифікації за короткий проміжок часу. Отримані впродовж виконання дипломної роботи результати дозволяють припустити можливість використання запропонованої архітектури для подальшої розробки стартапу. Всі розрахунки повинні виконуватись на стороні серверу, для того щоб не навантажувати прилад користувача, а також для того щоб не залежати від обчислювальної потужності обраного пристрою користувача. Таким чином, основні розрахунки будуть виконуватись на сервері який оброблятиме HTTP запити від користувача, а інтерфейс може реалізовуватись для браузерів або мобільних додатків. Оскільки популярність використання смартфонів в останні роки зростає, для початку пропонується розробити мобільний додаток, хоча базова версія додатку була створена для веб-застосунків, все ж більш домінуючою сферою є якраз використання мобільних додатків.

Для реалізації стартапу необхідна ідея, і коли вона готова, для успішності проекту, потрібно оцінити всі ризики і загрози, порівняти майбутній продукт з розробками конкурентів, оцінити його слабкі і сильні сторони, розглянути потенційних покупців та оцінити, наскільки ринок готовий до такого продукту. Саме це і буде описано в цьому розділі.

6.1 Опис ідеї проекту

Будь-який проект починається з ідеї. Розробка даної системи – не виняток. У таблиці 6.1 описана ідея розробки стартап-проекту.

Таблиця 6.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Додаток класифікації звучання музичних інструментів для мобільних пристроїв	Класифікація звучання музичних інструментів для створення аудіо композицій	Визначення музичного інструменту, що звучить в аудіо-записі
	Пост-продакшн професіонали, музиканти, аудіо-інженери, звукові студії	Дозволяє зекономити час завдяки використанню простого і зрозумілого інтерфейсу

Як видно з таблиці вище, для розробленого продукту напрямками застосування обрано користувачів, які працюють з обробкою та аналізом звучання музичних інструментів, а також люди які не мають достатньої кваліфікації в сфері музики, однак потребують класифікації музичних інструментів за фрагментом аудіо-запису. Розробка продукту як мобільного застосунку, є вигідною для користувачів, тому що їм не потрібно переносити обладнання чи використовувати власний комп'ютер для класифікації звуків. Достатньо простого мобільного пристрою з доступом до інтернету, який зазвичай є під рукою у кожної людини.

Розробка інтерфейсу окремо від серверної частини зробить можливим створення різних застосунків, наприклад мобільних додатків та веб-сторінок, що працюватимуть поверх однієї системи класифікації аудіо-записів, і допоможе зекономити ресурси на розробку окремих платформ, що працюватимуть на різних типах пристроїв. Аналіз даних який виконується на стороні серверу також покращить взаємодію користувача з додатком, оскільки йому не потрібно буде встановлювати додаткове програмне забезпечення, що буде даватиме змогу працювати з класифікатором на основу нейронної мережі.

Наступним кроком є визначення характеристик продукту, що планується розробити, а також порівняння розроблюваного продукту з характеристиками потенційних конкурентів, що уже є на ринку або планують свій запуск найближчим часом. В результаті потрібно виділити слабкі, нейтральні та сильні сторони продукту, за допомогою аналізу таблиці.

Таблиця 6.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Розроблюваний проект	Конкурент 1	Конкурент 2	Конкурент 3			
1	Точність	середня	середня	висока	низька		+	
2	Собівартість	низька	середня	висока	низька			+
3	Використання додаткового обладнання	Ні	Ні	Так	Так			+
4	Кросплатформенність	Ні	Ні	Так	Ні	+		

З таблиці вище зрозуміло, що відбувається порівняння з деякими конкурентами. Конкурент 1 – Musition Instrument Recognition від Rising Software, 2 – Audionamix, 3 – 3Strike Instruments.

Аналізуючи цю таблицю, стає зрозуміло, що сильними сторонами розроблюваного продукту є відсутність використання додаткового обладнання, а також низька ціна. На жаль, продукт не є кросплатформним, що є його слабкістю. Слід додати, що програма має непогану точність, якої вистачить для якісної класифікації відомих музичних інструментів.

На основі розглянутих міркувань та аналогів можна стверджувати конкурентну спроможність застосунку.

6.2 Технологічний аудит ідеї проекту

В цьому пункті слід провести аудит технології проекту, що має допомогти реалізувати ідею (технології створення продукту).

Перший крок для цього – це визначення технологій, які можуть використовуватися в процесі розробки (таблиця 6.3).

Таблиця 6.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Бібліотеки зразків звучання музичних інструментів	London Philharmonic Orchestra dataset	Наявна	Доступна
		IRMAS	Наявна	Доступна
		Musical instrument sample	Наявна	Доступна
		Vienna Symphonic Library	Наявна	Платна
2	Засоби розробки нейронних мереж	TensorFlow	Наявна	Доступна
		Caffe2	Наявна	Доступна
		Microsoft ML	Наявна	Доступна
3	Засоби екстракції музичних характеристик	Aubio	Наявна	Доступна
		LibROSA	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: London Philharmonic Orchestra dataset, TensorFlow та LibROSA				

З наведеної таблиці видно, що для розробки програмного забезпечення існують подібні аналоги, і вибір засобу може змінюватись при необхідності без значних втрат. Однак, обрані програмні засоби дозволяють забезпечити усі необхідні інструменти для розробки класифікатору на основі нейронних мереж.

Для створення нейронної мережі було обрано фреймворк TensorFlow через його підтримку компанією Google, великою популярністю серед розробників нейронних мереж та великою кількістю прикладів використання даного інструменту розробки.

Для обробки аудіосигналів та екстракції характеристик обрано бібліотеку LibROSA, яка надає простий та зручний інтерфейс для виокремлення найнеобхідніших звукових характеристик.

Для бази тренувальної вибірки було обрано набір аудіо-записів звучання музичних інструментів Лондонського симфонічного оркестру, в ньому представлено багато інструментів з різних груп, для зі звучанням в різних тонах, нотах та стилістикою виконання.

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час впровадження проекту для користувачі, та загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Отже, дані характеристики вкажуть на можливе місце нового застосунку на ринку, його потенційних користувачів, прогноз розвитку та рентабельність.

Спочатку варто провести аналіз попиту (таблиця 6.4).

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Біля 5
2	Загальний обсяг продаж, грн/ум.од	10 000 грн за ум.од.
3	Динаміка ринку (якісна оцінка)	Очікується зростання
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	15

Аналізуючи попередню таблицю, зрозуміло, що вкладання коштів в цей проект є вигідним. Слід звернути увагу, що обмежень для входу і до стандартизації немає, що полегшує ситуацію. До того ж, на ринку очікується зростання в цій сфері, оскільки в останні роки інтерес до використання та розробки штучного інтелекту для обробки аудіосигналів та систем пов'язаних з його методами тільки зростає.

Наступним кроком є визначення цільової аудиторії програмного продукту, її поведінки та вимог (таблиця 6.5). Таблиця дозволить проаналізувати потреби ринку та цільову аудиторію, яка може бути зацікавлена в виході проекту.

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінність у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Можливість швидко і зручно здійснювати класифікацію звучання музичних інструментів	Музиканти	Не передбачається	Зручність інтерфейсу, швидкість роботи, можливість швидко класифікакувати музичний інструмент за звучанням
2		Пост-продакшн професіонали, аудіо-інженери, звукові студії	Не передбачається	Зручність інтерфейсу, швидкість роботи, можливість швидко класифікакувати музичний інструмент за звучанням

Як видно з попередньої таблиці, цільовою аудиторією є пост-продакшн професіонали, музиканти, аудіо-інженери та працівники звукових студій. При цьому відмінності в поведінці цих двох цільових аудиторій не очікується: всі вони бажають одного і того ж – швидкої та якісної класифікації звучання музичних інструментів. Щоправда, вимоги у пост-продакшн професіоналів, аудіо-інженерів та працівників звукових студій можуть бути трошки вищі, адже від цього залежить результат їхньої професійної діяльності.

Наступним кроком є аналіз ринкового середовища. Необхідно в табличному вигляді визначити фактори загроз (таблиця 6.6) і можливостей (таблиця 6.7), що перешкоджають і допомагають впровадженню програми.

Таблиця 6.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Ризик технології	Розпізнавання без використання додаткового обладнання згодом може стати недостатньо точною для певного прошарку клієнтів	Підвищення точності за рахунок введення якого одного типу додаткових засобів (наприклад, Wi-Fi)
2	Недоступність платформи	Неможливість використання програмного забезпечення на пристроях, що не запускаються на мобільній платформі	Розробка програмного забезпечення для інших платформ (наприклад, iOS чи web-застосунок).

В таблиці було наведено основні загрози виходу на ринку розроблюваного продукту. На думку розробників, найголовнішою з них є ризик технології.

Так, раніше згадувалося, що відмова від використання додаткового обладнання, з однієї сторони, сприяє низькій собівартості продукту і легкості його розгортання. З іншого боку, для деяких користувачів це може означати недостатню точність класифікації, що зменшить коло потенційних користувачів. Якщо таке траплятиметься часто, команді розробників варто, наприклад, удосконалити свій продукт, використавши якийсь один тип додаткових засобів.

Також існує загроза, що програний продукт не буде запускатися на платформах, що працюють не на мобільній платформі. Оскільки на першому етапі планується запровадити продукт на мобільну платформу android, то потенційно можна втратити користувачів, які використовують інші мобільні платформи або веб-застосунки. В такому випадку є можливим варіант розроблення такого ж програмного забезпечення, але вже підлаштованого під інші мобільні платформи або веб-браузери. Можливим вирішенням є випуск окремого модуля, який зможе працювати з класифікатором звучання гри музичних інструментів і буде мати низьку собівартість.

Таблиця 6.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Розвиток технологій для розпізнавання звучання музичних інструментів, популяризація класифікаторів для навчання дітей у школах.	Існує можливість, що при збільшенні потенційних покупців (структур, яким може знадобитися продукт), збільшить і попит на забезпечення	Посилення рекламної кампанії

Найголовнішим фактором можливостей є популяризація технологій штучного інтелекту в школах, що може збільшити кількість користувачів. Також варто врахувати, що розвиток дозволить готувати музикантів, або тренувати їхній слух на розпізнавання звучання певних музичних інструментів. Запровадження технологій в цих сферах може надати сильний поштовх до розвитку подібних систем і забезпечити більше фінансування.

Наступним кроком є визначення типу конкуренції (таблиця 6.8).

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика
Тип конкуренції	Олігополія
За рівнем конкурентної боротьби	Глобальна
За галузевою ознакою	Міжгалузева
Конкуренція за видами товарів	Товарно-видова
За характером конкурентних переваг	Нецінова
За інтенсивністю	Не марочна

Такий аналіз конкуренції є непоганим, але неповним. Оскільки не містить детального опису характеристик. Саме тому варто провести аналіз конкуренції в галузі за М. Портером (таблиця 6.9).

Таблиця 6.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальник	Клієнти	Товари-замінники
	Rising Software Audionamix 3Strike Instruments	Наявність аналізу аудіо-запису, додаткове обладнання	Інтернет ресурси	Мобільні застосунки	відсутні
Висновки :	Існує близько 3 найвпливовіших конкурентів. Конкурент 2 (Audionamix) є найбільш серйозним, адже використовує схожий підхід і технології	Можливості виходу на ринок є, але вони також є у потенційних конкурентів. Строки виходу їх продукту на ринок — приблизно 1 рік	Постачальник и відсутні	Зручність, достатня точність	Загроза розробки конкурентами більш точного та (або) дешевшого аналогу

Попередня таблиця нам дає зрозуміти, що продукт має одного найсерйознішого конкурента, що має аналогічний підхід рішення ситуації. Врахувавши зручність і точність розроблюваного програмного продукту, а також відмову від використання додаткових пристроїв, можна вважати, що він має можливість роботи на ринку.

Використовуючи ці висновки, можна виокремити основні фактори конкурентоспроможності продукту (таблиця 6.10).

Таблиця 6.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Функціонал	Наявність режиму швидкого розпізнавання за допомогою обробки даних на сервері. Можливість обробки сигналів на льоту, без збереження даних в додатку користувача

Таблиця 6.10 (продовження)

2	Вартість	Передбачається випуск товару з ціною, нижчою ніж у конкурентів
3	Не потребує додаткового обладнання	Для роботи програми не потрібні, наприклад, синтезатори

З таблиці ми бачимо, що відмова від додаткового обладнання, режим з розпізнавання на льоту і невисока вартість продукту (у порівнянні з конкурентами) є основними факторами, що дають перевагу над конкурентами.

Після таких висновків вже можна проаналізувати сильні та слабкі сторони проекту у порівнянні з конкурентами (таблиця 6.11).

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентноспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розроблюваним продуктом						
			-3	-2	-1	0	1	2	3
1	Функціонал	17			+				
2	Вартість	19	+						
3	Не потребує дод. обладнання	17			+				

Посилаючись на цю таблицю, ми бачимо, що найбільше балів набрано за фактором "вартість". В двох інших категоріях розроблюваний програмний продукт також має перевагу.

Заключною таблицею ринкового аналізу можливостей впровадження розробленого програмного продукту є аналіз SWOT, що представляє собою сильні та слабкі сторони, можливості і загрози для проекту в одній таблиці (таблиця 6.12).

Таблиця 6.12. SWOT-аналіз стартап-проекту

Сильні сторони: функціонал, вартість, не потребує додаткового обладнання	Слабкі сторони: Недоступність на платформах, що не підтримують нейронні мережі, недостатня точність для деяких клієнтів
--	---

Таблиця 6.12 (продовження)

Можливості: Продаж більшої кількості товару через збільшення навчальних закладів Індивідуалізація програми під конкретного замовника Гібридизація технологій для підвищення точності	Загрози: віддання переваги потенціальними клієнтами продуктам конкурентів через недоступність інших платформ та технологічний ризик (не використовуються дод. апаратура)
---	--

Вищенаведена таблиця по суті є узагальненням вищесказаного. Тут варто додатково пояснити тільки кілька важливих моментів.

Якщо клієнти віддаватимуть перевагу продуктам конкурентів через недостатню точність (техн. ризик), то, можливо, варто впровадити гібридизацію технологій, що означає залучення якогось одного типу дод. обладнання до роботи (наприклад, Wi-Fi).

Індивідуалізація програми під конкретного замовника може бути виконана, наприклад, тренування класифікатору на базі даних користувача або додавання класифікації звучання гри специфічного музичного інструменту.

Використовуючи SWOT-аналіз, можна скласти таблицю 6.13 альтернатив ринкового впровадження стартап-проекту.

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива	Ймовірність отримання ресурсів	Строки реалізації
1.	Розробка системи класифікації звучання з використання додаткового обладнання (синтезатори)	20%	1,5
2.	Розробка системи класифікації звучання в реальному часі	80%	1

Враховуючи вищенаведену таблицю, зрозуміло, що для реалізації стартап-проекту доцільніше взяти другу альтернативу, а саме розробку системи класифікації звучання в реальному часі, адже, по-перше, на це піде менше часу, по-друге, не слід забувати, що стартап-проект є ще і дипломною роботою, тому при обранні першої альтернативи очікуються проблеми з наданням коштів для додаткового обладнання, тому раціональніше обрати інший спосіб.

6.4 Розробка ринкової стратегії проекту

В розробленні ринкової стратегії перше, що треба зробити, це визначити стратегію охоплення ринку (таблиця 6.14).

Таблиця 6.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Музиканти	+	Середній	Практично немає	Просто
2	Музичні заклади	+	Високий	Вище середнього	Вище середнього
3	Звукові студії та кіноіндустрія	+	Вище середнього	Висока	Складно
Які цільові групи обрано: 1, 2					

Переглядаючи таблицю вище, зрозуміло, що за цільові групи користувачів обрано музикантів та музичні заклади. Хоч попит є у всіх трьох категорій, слід зауважити, що через відсутність конкуренції в сфері музикантів цю категорію найпростіше охопити. Також можна спробувати запропонувати даний продукт музичним закладах, хоч там і досить велика конкуренція. Для останньої категорії даний продукт підходить не зовсім: для професійних цілей треба продукт, що має більшу точність.

На основі цих міркувань можна навести базову стратегію розвитку, яка покаже на кроки, які потрібно реалізувати і яким шляхом краще слідувати, щоб досягти хорошого результату(таблиця 6.15).

Таблиця 6.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розробка системи класифікації з використанням тільки штучного інтелекту	Стратегія диференційованого маркетингу	Функціонал, Вартість, Не потребує додаткового обладнання	диференціація

Так як планується співпраця з кількома сегментами ринку, за стратегію охоплення ринку було обрано стратегію диференційованого маркетингу; до того ж, деякі особливості продукту будуть змінюватись в залежності від сегменту і конкретного користувача. Базовою стратегією розвитку визначено диференціацію, адже товар пропонує не зовсім поширений підхід: доповнену реальність і відмову від додаткового обладнання.

Наступний крок – визначення базової стратегії конкурентної поведінки (таблиця 6.16)

Таблиця 6.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект "першопрохідцем" на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	Шукати нових споживачів	Так	Стратегія наслідування лідеру

Звісно, розроблений продукт не буде першопрохідцем на ринку: до нього вже існували системи класифікації звучання музичних інструментів; дуже рідко, але деякі навіть використовували онлайн класифікацію. Найоптимальнішим варіантом буде пошук зовсім нових користувачів (наприклад, ті ж музичні заклади). Неможливо створити зовсім нову систему класифікацію, відмінну від тих, що пропонують конкуренти, тому деякі характеристики будуть однакові.

Зважаючи на вимоги користувачів, стратегій розвитку та конкурентної поведінки, далі треба розробити стратегію позиціонування. Під цією стратегією розуміється формування ринкової позиції (комплексу асоціацій), що допоможе споживачам з ідентифікацією проекту (таблиця 6.17).

Таблиця 6.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувану комплексну позицію власного проекту (три ключових)
1	Дешевизна, зручність інтерфейсу, точність	Диференціація	Відмова від додаткового обладнання зробить продукт дешевшим, а режим класифікації на льоту, без попереднього збереження допоможе звучніше для користувача і швидше розпізнати звучання інструменту	Доступність, зручність, функціональність

Попередня таблиця нам яскраво демонструє комплексну позицію проекту, а саме: доступність, зручність, і функціональність, що сформована, виходячи з вимог користувачів з обраних сегментів і ключових конкурентоспроможних позицій проекту.

6.5 Розроблення маркетингової програми

Розробляючи маркетингову програму, з самого початку потрібно сформувати маркетингову концепцію товару (таблиця 6.18).

Таблиця 6.18. Визначення ключових переваг концепції потенційного товару

п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Функціонал	Режим розпізнавання на льоту	Наявність режиму, якого практично немає ні в одного з конкурентів
2	Вартість	Вигідна ціна для покупців	Нижча ціна, ніж у конкурентів
3	Незалежність системи від дод. пристроїв	Не потребує додаткового обладнання	Потрібен тільки смартфон

Наступний крок – розробка трирівневої маркетингової моделі товару.

Таблиця 6.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Додаток для смартфона, що допомагає здійснювати розпізнавання звучання музичних інструментів використовуючи штучний інтелект		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Низька ціна	Нм	Тх
	Функціонал		
	Не потребує дод. обл.		
	Зручний інтерфейс		
	Якість: продукт протестовано вручну		
Маркування: Немає			
Компанія: назва “RegocInc”			
III. Товар із підкріпленням	Продаж товару за підпискою; постійна підтримка придбаного додатку		
За рахунок чого потенційний товар буде захищено від копіювання: обфускація коду			

В попередній таблиці було наведено три рівні моделі товару. Це мобільний додаток, що пропонується для полегшення класифікації звучання музичного інструменту. Слід зауважити, що програма не купується тільки один раз, а доступна за підпискою. Це у свою чергу означає постійну і якісну підтримку придбаного програмного продукту. Товар буде захищено від копіювання та неправомірного розповсюдження за допомогою обфускації коду.

Далі потрібно визначити цінові межі (таблиця 6.20).

Таблиця 6.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар
1.	15000	25000	35000	5000-10000

Після цього необхідно визначити систему збуту, оптимальну для даного проекту, яка допоможе зрозуміти як краще реалізовувати продукт, за допомогою посередників за власними ресурсами (таблиця 6.21).

Таблиця 6.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Покупка та щомісячна підписка	Продаж	0 (напрямую)	Власна

Попередня таблиця презентує систему збуту. В її рамках прийняті такі рішення, що користувачі купуватимуть щомісячну підписку і що система збуту буде без посередників, власна, тобто глибина каналу збуту – 0.

Завершальним етапом формування маркетингової програми є розробка концепцій маркетингових комунікацій (таблиця 6.22).

Таблиця 6.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення
1.	Завантаження додатку з плей маркету.	Інтернет	Низька ціна, функціонал, не потрібно додаткового обладнання, зручний інтерфейс	Зацікавити потенційних користувачів; довести, що розроблений додаток економить час та допомагає спростити процес розпізнавання.

Посилаючись на попередню таблицю, що показує концепцію маркетингових комунікацій для розробленого програмного продукту, зрозуміло, що додаток можна буде завантажити з плей маркету, і авторизуватись у свій профіль.

Висновки до розділу 6

В рамках розділу були розглянуті головні моменти виходу на ринок запропонованого програмного продукту, що є мобільним додатком розпізнавання звучання музичних характеристик з використанням штучного інтелекту.

Було запропоновано та детально розглянуто ідею для проекту, а також проведено технологічний аудит та аналіз ринкових можливостей запуску стартап-проекту. Після цього наступним кроком було розроблено ринкову стратегію проекту. В кінці розділу було розроблено маркетингову програму для стартапу.

Для ринкової реалізації проекту було обрано альтернативу з використанням розпізнавання звучання музичного інструменту на льоту, без додаткового збереження запису.

Крім того, було проаналізовано слабкі та сильні сторони проекту. До перших можна сміливо віднести відмову від використання додаткового обладнання, використання штучного інтелекту та низьку цінову пропозицію, а до останніх – підтримка не на всіх смартфонах та недостатня точність для деяких користувачів.

Було проаналізовано можливі ризики та запропоновано необхідні дії в разі їх прояву, в тому числі і гібридизація технології при технологічному ризику. Враховуючи це, було розроблено SWOT-аналіз.

В ході розробки програми було детально порівняно розроблюваний проект з аналогічними програмами конкурентів, такими як Musition Instrument Recognition від Rising Software, Audionamix та 3Strike Instruments. Порівняльний аналіз показав, що і в конкурентів, і в розроблюваній програмі є свої плюси та мінуси, проте в загальному остання має більш кращу і вигідну для користувачів сукупність показників, маючи таку комплексну позицію: доступність, функціональність і зручність.

Було виявлено, що існує можливість комерціалізації такого проекту, адже показники ринку цьому сприяють (наприклад, є попит на подібні товари, очікується зростання ринку у цій сфері). Враховуючи потенційні групи клієнтів, невисокі бар'єри входження, доведену вище конкурентоспроможність продукту, можна зробити висновок про існування перспектив впровадження такого проекту.

ВИСНОВКИ

1. Виконано огляд підходів до класифікації музичних інструментів засобами машинного навчання. Визначено, що на сьогодні найбільш ефективними в якості класифікаторів є нейронні мережі різної архітектури. Визначено, що розв'язання задачі складається з двох етапів: створення навчальної вибірки та створення класифікатору на основі нейронної мережі.

2. Розглянуто основні часові та спектральні характеристики звукового сигналу, що використовуються в задачі класифікації музичних інструментів. Для створення класифікатору в якості характеристик обрано мел-частотні кепстральні коефіцієнти (MFCC) та спектральний центроїд (spectral centroid).

3. Проведено аналіз існуючих програмних фреймворків машинного навчання. Обрано для подальшого використання відкриту програмну бібліотеку Tensorflow.

4. З метою створення навчальної вибірки в роботі виконано огляд основних репозиторіїв екземплярів запису музичних інструментів. Для екстракції характеристик обрано бібліотеку Лондонського симфонічного оркестру.

5. Для перевірки ефективності використання нейронних мереж в якості класифікатору музичних інструментів розроблено два класифікатори, один на основі нейронної мережі прямого поширення, а інший – згорткової нейронної мережі. Отримані результати склали 90,2% для нейронної мережі прямого поширення та 97,9% для згорткової.

На основі отриманих результатів можна стверджувати що ефективним вирішенням задачі аудіосигналів є наступний підхід: підбір значущих музичних характеристик, пошук бази даних, екстракція характеристик, створення архітектури класифікатору на основі нейронної мережі, розробка та тренування розробленої моделі. Слідування даним крокам значно спрощує процес розробки застосунків у сфері обробки аудіосигналів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Eronen A. AUTOMATIC MUSICAL INSTRUMENT RECOGNITION / Antti Eronen., 2001. – 74 с.
2. Han Y. Deep convolutional neural networks for predominant instrument recognition in polyphonic music / Y. Han, J. Kim, K. Lee. // IEEE. – 2016.
3. Singh P. Implementing Musical Instrument Recognition using CNN and SVM / P. Singh, B. Dnyaneshwar. // International Research Journal of Engineering and Technology. – 2019. – С. 1487–1493.
4. Toghiani-Rizi B. Musical Instrument Recognition Using Their Distinctive Characteristics / Babak Toghiani-Rizi. // Uppsala University. – 2016.
5. Scott S. An examination of evaluation algorithms for the RBF method, Engineering Analysis with Boundary Elements / S. Scott, S. Cogar. – 2017. – №75. – С. 36–45.
6. Self Organizing Maps [Електронний ресурс] // Towards Data Science. – 2018. – Режим доступу до ресурсу: <https://towardsdatascience.com/self-organizing-maps-ff5853a118d4>.
7. TensorFlow - Multi-Layer Perceptron Learning [Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/tensorflow/tensorflow_multi_layer_perceptron_learning.htm.
8. Toghiani-Rizi B. Musical Instrument Recognition Using Their Distinctive Characteristics / Babak Toghiani-Rizi. // Uppsala University. – 2017.
9. Convolutional Neural Network [Електронний ресурс] // Stanford – Режим доступу до ресурсу: <http://deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>.
10. Singh P. Implementing Musical Instrument Recognition using CNN and SVM / P. Singh, B. Dnyaneshwar. // International Research Journal of Engineering and Technology. – 2019. – С. 1487–1493.

11. Daeyeol K. A Single Predominant Instrument Recognition of Polyphonic Music Using CNN-based Timbre Analysis / Kim Daeyeol. // International Journal of Engineering & Technology. – 2018. – С. 589–593.
12. Mingqing Y. DEEP LEARNING FOR MUSICAL INSTRUMENT RECOGNITION / Yun Mingqing. // Department of Electrical and Computer Engineering University of Rochester. – 2017.
13. Yun-Ning H. FRAME-LEVEL INSTRUMENT RECOGNITION BY TIMBRE AND PITCH / Hung Yun-Ning. // Research Center for IT Innovation. – 2018. – С. 135–143.
14. IRMAS [Електронний ресурс] // Music technology group – Режим доступу до ресурсу: <https://www.upf.edu/web/mtg/irmas>.
15. Walker J. The Use of Mel-frequency Cepstral Coefficients in Musical Instrument Identification / Jacqueline Walker. // University of Limerick. – 2013.
16. Banchhor K. S. Musical Instrument Recognition using Zero Crossing Rate and Short - time E nergy / K. S. Banchhor, A. Khan. – New York: Foundation of Computer Sci ence FCS, 2012. – 4 с. – (ISSN). – (2249 - 0868).
17. Bachu R. G. 1 Separation of Voiced and Unvoiced using Zero crossi ng rate and Energy of the Speech Signal / Bachu. – 2005. – С. 7.
18. Eronen MSC [Електронний ресурс] – Режим доступу до ресурсу: http://www.cs.tut.fi/sgn/arg/music/eronen/eronen_msc.ps.
19. Ericsson L. Automatic speech/music discrimination in audio files / Lars Ericsson. // Royal Institute of Technology. – 2009. – С. 40.
20. Logan B. Mel-frequency cepstral coefficients for music modelling / Beth Logan., 2000. – 13 с.
21. Brown J. Calculation of a constant Q spectral transform / Judith Brown. – Массачусетс: Массачусетський Інститут Технологій, 1990. – 10 с.
22. On the relevance of spectral features for instrument classification / A.Nielsen, S. Sigurdsson, L. Hansen, J. Arenas-García., 2007. – 5 с. – (IEEE).
23. Google trends [Електронний ресурс] – Режим доступу до ресурсу: <https://trends.google.com/trends/explore?date=2016-01-01%202018-06-03&q=machine%20learning>.

24. Caffe2 Operators Overview [Электронный ресурс] – Режим доступа до ресурсу: <https://caffe2.ai/docs/operators.html>.
25. Apache SINGA documentation [Электронный ресурс] – Режим доступа до ресурсу: http://singa.apache.org/en/docs/software_stack.html.
26. Windows ML Architecture [Электронный ресурс] – Режим доступа до ресурсу: <https://mspoweruser.com/microsoft-announces-windows-ai-platform/windows-ml-architecture/>.
27. MXNet System Architecture [Электронный ресурс] – Режим доступа до ресурсу: <https://mxnet.incubator.apache.org/architecture/overview.html>.
28. Torch documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://github.com/soumith/cvpr2015/blob/master/cvpr-torch.pdf>.
29. Interest around deep learning frameworks [Электронный ресурс] – Режим доступа до ресурсу: <https://twitter.com/fchollet/status/871089784898310144>.
30. Theano Group [Электронный ресурс] – Режим доступа до ресурсу: 1. <https://groups.google.com/forum/#!topic/theano-users/7Poq8BZutbY>.
31. Introduction to TensorFlow Datasets and Estimators [Электронный ресурс] – Режим доступа до ресурсу: <https://developers.googleblog.com/2017/09/introducing-tensorflow-datasets.html>.
32. Aubio documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://aubio.org/>.
33. Yaafe documentation [Электронный ресурс] – Режим доступа до ресурсу: <http://yaafe.sourceforge.net/>.

ДОДАТОК А

АРХІТЕКТУРА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ОБРОБКИ АУДІОСИГНАЛІВ

Тези на конференцію

«Сучасні проблеми наукового забезпечення енергетики»

УКР.НТУУ"КПІ ім. Ігоря Сікорського" _ТЕФ_АПЕПС_ ТВ4194_19М

Аркушів 1

Київ 2019

АРХІТЕКТУРА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ОБРОБКИ АУДІОСИГНАЛІВ

Автоматизація процесів обробки аудіосигналів вимагає вирішення задач ідентифікації, класифікації та розпізнання джерел звукової інформації. Одним зі світових трендів вирішення таких типів задач на сьогодні є нейронні мережі. В той же час, створення подібних систем потребує від розробника знань не лише з області обробки аудіоданих, але й машинного навчання. Актуальною є проблема навчання спеціалістів, здатних застосовувати сучасні фреймворки розробки нейронних мереж для вирішення задач обробки звукових сигналів.

Зазвичай, для демонстрації базових підходів та порівняння ефективності фреймворків поширюються набори прикладів з вирішення основних класів задач. Для нейронних мереж історично найчастіше використовуються задачі обробки візуальної інформації. В той же час розробка загальних рішень, які поєднують в собі основні етапи реалізації нейронних мереж для класифікації аудіосигналів, має практичне значення як з точки зору навчання спеціалістів, так і в якості засобу демонстрації можливостей програмного засобу.

Метою даної роботи є розробка архітектурного рішення та реалізація класифікатору музичних інструментів з використанням фреймворку розробки нейронних мереж та бібліотек обробки звукової інформації. Задача розпізнавання музичних інструментів була обрана як така, що може представляти інтерес в якості прикладу та, в той же час, потребує вирішення в складних підсистемах пошуку та аналізу музичної інформації.

Вирішення задачі розпізнавання передбачає наступні етапи: екстракція характеристик аудіозапису; виокремлення значимих характеристичних ознак; формування навчальної вибірки; розробка архітектури нейронної мережі; навчання класифікатору. Вищенаведені етапи визначають загальну схему навчання та використання моделі класифікатору. В режимі навчання на вхід застосунку подається множина аудіофайлів з відомими класами музичних інструментів. Формування навчальної вибірки передбачає екстракцію попередньо обраних значимих характеристик звукового сигналу. Виокремленні характеристики надходять на вхід алгоритму навчання нейронної мережі. Вихідними даними цього етапу є модель нейронної мережі, яка власне і виступає класифікатором. В режимі класифікації на вхід системи подається аудіофайл запису невідомого заздалегідь музичного інструменту. Даний аудіофайл проходить етапи екстракції характеристик, аналогічно до файлів навчальної вибірки. Вектор характеристик подається на вхід до класифікатору. На виході отримується припущення щодо музичного інструменту та імовірність, яка показує впевненість класифікатора у своєму рішенні.

Кожен з етапів розпізнавання передбачає вибір засобів реалізації, які задовольняють вимогам ефективності та простоти інтеграції в загальне рішення.

В класифікації звучання музичних інструментів можна виокремити два типи підзадач: розпізнавання за моно-звуком та за поліфонією. В даній роботі запропоновано виконувати ідентифікацію за записом моно пасажу з декількох нот. Для створення класифікатору в якості характеристик обрано мел-частотні кепстральні коефіцієнти (MFCC) та додаткові характеристики, а саме: частота згортання (roll-off), спектральний центроїд (spectral centroid) та тональність (tonnetz).

Реалізацію модулю екстракції характеристик було виконано з використанням бібліотеки LibROSA [1]. Для виокремлення значимих характеристик було використано алгоритм Principal component analysis (PCA). Для його реалізації застосовано бібліотеку Keras.

З метою формування тренувальної вибірки було проведено огляд репозиторіїв аудіозаписів інструментів, що поширюються за вільною ліцензією. На основі порівняльного аналізу обрано бібліотеку Лондонського симфонічного оркестру [2].

Для реалізації нейронної мережі обрано засоби розробки бібліотеки TensorFlow [3]. Запропоновано архітектуру тришарової нейронної мережі прямого поширення. Мережа містить два приховані (hidden) шари по 16 вузлів кожен. Експериментально функцією активації було обрано ReLU, оптимізатором - RMSProp. Вихідний шар містить 8 нейронів, відповідно до кількості класів музичних інструментів, які розглянуто в роботі

Рішення було апробовано на задачі розпізнавання духових та струнних музичних

інструментів. За результатами експериментів було отримано середню точність розпізнавання 89,43%.

Таким чином, в даній роботі запропоновано архітектуру застосунку, яка передбачає використання бібліотек обробки звукових сигналів в поєднанні з фреймворками машинного навчання. Обрані спеціалізовані програмні засоби істотно спрощують процес розробки та дозволяють досягти результату на рівні класичних алгоритмів класифікації.

Запропоноване рішення може бути використане як ілюстративний матеріал в процесі навчання або ж як архітектурне рішення в подальших розробках.

Перелік посилань:

1. Librosa. A python package for music and audio analysis. [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://github.com/librosa/librosa>.
2. L. P. Orchestra. Sound samples. [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: http://www.philharmonia.co.uk/explore/sound_samples.
3. TensorFlow. An open-source machine learning framework for everyone. [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://www.tensorflow.org/>